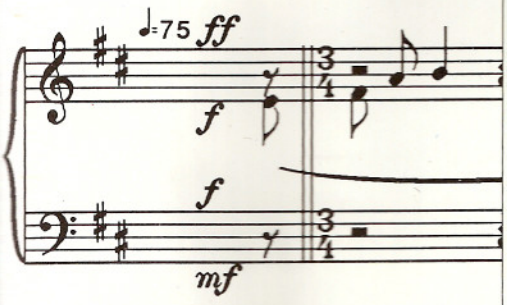
La revue francophone indépendante pour les utilisateurs d'Apple

POM'S



Essai ConcertWare

☞

Un CDA GS

❑

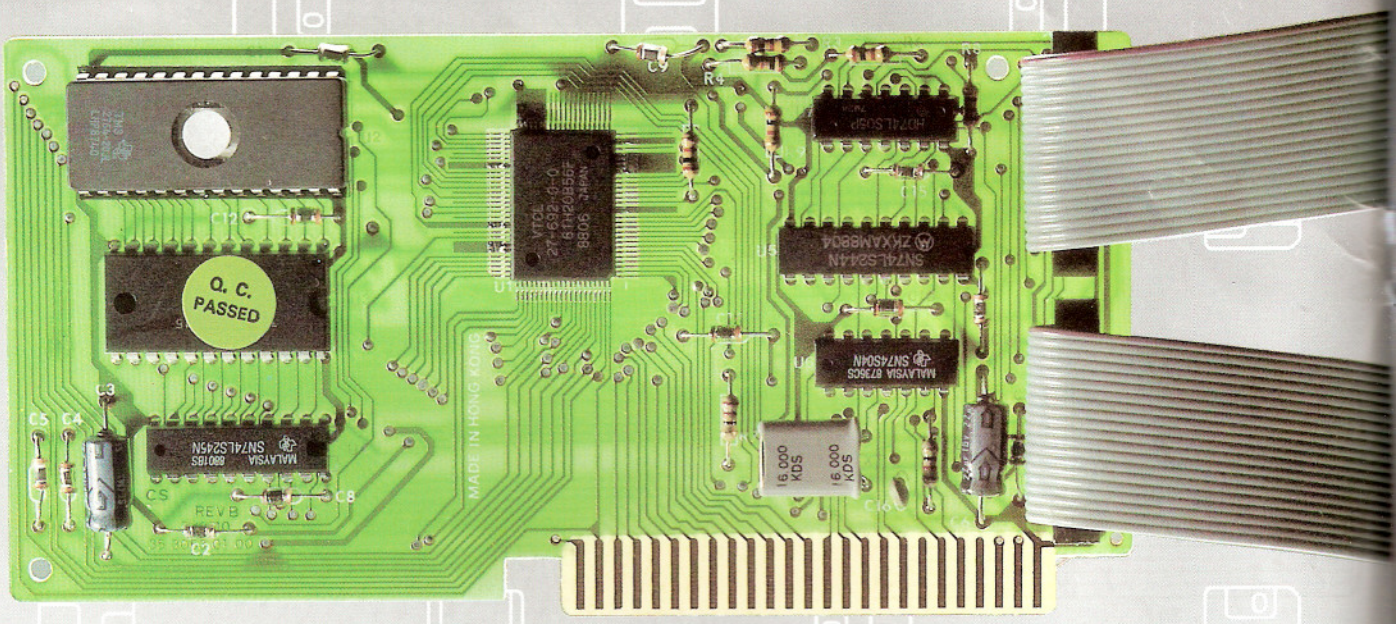
Stricto Sensu, accessoire Mac

❑

Macros AppleWorks

M 2366 - 44 - 45,00 F
3792366045005 00440

Universal Disk Controller



...LA solution pour relier aux Apple //e et][+ :

- un "Apple 3.5 GS Platinum" 800Ko
- un lecteur Macintosh 800Ko
- un lecteur Macintosh 400Ko
- un lecteur Chinon 800Ko
- un Unidisk 5,25" 140Ko

Connectée à n'importe quel port, elle pilote deux lecteurs (même différents, pouvant totaliser 1,6 méga). Elle permet le travail bit à bit avec tous les lecteurs d'aujourd'hui (ce travail était impossible avec l'Unidisk 3.5").

Carte UDC : 1 050,00 F TTC franco
Tarif carte Pom's Joker : 945,00 F TTC

Logma S.A. – 12, rue d'Anjou – 78000 Versailles – Tél : 39 51 24 43

Numéro 44
septembre-octobre 1989

Éditorial
Hervé Thiriez
Page 7

GS-Init
Jean-Yves Bourdin
Page 8

Les nouveaux TimeOut
Dimitri Geystor
Page 15

Programmation AppleWorks
Des Ascii et des
Dimitri Geystor
Page 18

Un nouveau CDA :
Ram.Use
Patrice Neveu
Page 23

Une commande externe :
ShortP
Gisèle Perreault
Page 29

GS & fenêtres :
Scrolling
Patrick Desnoues
Page 32

Graphisme et son :
DPA 23, 24, 25, 26
Page 40

Un accessoire Mac :
Stricto Sensu
Jean-Luc Bazanegue
Page 42

À l'essai :
ConcertWare
Christian Piard
Page 47

ProCommand
Pierre Demblon
Page 36

Macintosh :
Les News
Ariel Sebban
Page 51

Apple // for ever
Jean-Yves Bourdin
Page 58

Petites annonces
Page 74

Mots croisés
Joëlle Piard
Page 74

Les annonceurs : Apple : pages 4 & 5 ; Bréjoux : page 25 ; Logitech : page 76.

Éditions MEV - 12, rue d'Anjou - 78000 Versailles. Tél. : (1) 39 51 24 43. Directeur de la publication : Hervé Thiriez

Pour savoir si la créativité a vraiment quelque chose à voir avec la productivité, venez à Apple Expo retrouver Apple et tous ses partenaires. Installée au CNIT, le plus grand centre de communication et d'échanges du monde, Apple Expo vous présente tous les moyens de développer la créativité et donc la productivité des hommes.

C'est l'occasion ou jamais de venir découvrir ou redécouvrir le « monde Macintosh ». Les nouveaux Macintosh vous seront présentés en exclusivité ainsi que les dernières applications et matériels périphériques vous permettant de développer toutes les facettes de votre créativité quel que soit votre domaine : traitements de texte et tableurs bien sûr, mais aussi publication et présentation assistées par ordinateur ou encore CAO, DAO. Enfin, vous verrez qu'il est très facile de mettre Macintosh en réseau ainsi

Apple, le logo Apple, Macintosh sont des marques déposées de Apple Computer Inc.

au service de la productivité.
voir.

CLM/BBDO

que d'entrer en contact avec d'autres environnements ; il n'existe donc plus de barrière au bureau, toute communication avec l'extérieur étant désormais très simple. Maintenant si vous désirez de plus amples informations, venez donc nous voir :

Apple Expo a lieu du 27 au 30 septembre au CNIT. Et si vous croisez cet homme dans sa grosse voiture, n'hésitez pas à lui en toucher deux mots. Ça ne peut que l'intéresser.

Pour recevoir votre invitation ou connaître le calendrier des conférences, tapez sur votre Minitel 3614 Code Apple.

Apple Expo

ExtraFonts

de Daniel Lurot

Deux applications, **ColorFonts** et **EdiFonts**, qui donneront a votre AppleWorks/TimeOut encore plus de souplesse et de puissance

- **ColorFonts** donne à TimeOut SuperFonts la **couleur**.
- **EdiFonts** est un éditeur de polices qui dépasse de loin les possibilités d'Edipol et FontViewer réunis.

ColorFonts

Votre SuperFonts monochrome (version française Pom's) est transformé en **ColorFonts**, permettant désormais l'impression dans les sept couleurs de l'Imagewriter II (**noir, jaune, rouge, bleu, orange, vert et violet**), en conservant bien sûr toutes les possibilités de SuperFonts (polices et styles variés, justification à droite, tabulations, impression d'images).
Les changements de couleurs peuvent, comme les changements de polices ou de styles, intervenir à tout endroit, même au milieu d'un mot. Vous pouvez aussi imprimer une **image** dans la **couleur** souhaitée.

En bonus, ColorFonts vous donne aussi accès aux caractères normalement inaccessibles par SuperFonts (codes 0 à 31).

Bien entendu, les commandes habituelles restent inchangées.

ColorFonts est compatible avec les autres imprimantes, mais imprime alors en monochrome.

EdiFonts

Une nouvelle application TimeOut

Avec EdiFonts, vous pourrez visualiser, modifier et imprimer vos polices de caractères, mais aussi en créer de nouvelles (en partant de zéro), copier des caractères d'une police vers une autre. Tout ceci sans quitter AppleWorks !
Avec à votre disposition un index des polices (analogue à l'index des fichiers AppleWorks) permettant de travailler sur plusieurs polices (jusqu'à sept à la fois).

Affichage de la police page par page, commandes **ergonomiques**

Accès rapide aux caractères désirés

Copie d'**écran graphique**

Impression de la police complète

Visualisation individuelle et **modification** des caractères

Édition **rapide** grâce à des commandes puissantes : insertion ou effacement de colonnes ou de lignes — **déplacement** du caractère tout entier dans les quatre directions — effacement de la grille d'édition — modification des 'width' et 'offset'

Création de **nouveaux** caractères

Copie de caractères interne, vers le presse-papiers, du presse-papiers

Suppression de caractères

Visualisation, impression et modification des **paramètres globaux** de la police — firstC — lastC — leading et kernMax — ascent — descent — nom de la police

---

La disquette ExtraFonts existe en 5,25"/140Ko et 3,5"/800Ko. ExtraFonts est compatible avec les Apple //e 128Ko, //c, IIGS.
ColorFonts nécessite AppleWorks, une ImageWriter II et TimeOut SuperFonts en version française.
EdiFonts nécessite AppleWorks et au moins un TimeOut en version française.

ExtraFonts : 500,00 F — Carte Joker : 450,00 F — Bon de commande page 70

Ont collaboré à ce numéro
Jean-Luc Bazanegue - Jean-Yves Bourdin
Pierre Demblon - Patrick Desnoues
Dimitri Geystor - Olivier Herz
Nathalie Laudat - Daniel Lurot
Gérard Michel - Patrice Neveu
Gisèle Perreault - Christian Piard
Joëlle Piard - Ariel Sebban
Hervé Thiriez

Directeur de la publication
rédacteur en chef
Hervé Thiriez

Rédacteurs
Jean-Luc Bazanegue - Christian Piard

Siège social
Éditions MEV – 12, rue d'Anjou
78000 Versailles – ☎ (1) 39 51 24 43
Serveur Minitel (1) 39 53 04 40

Publicité
Éditions MEV

Diffusion
N.M.P.P.

Impression
MAME — 49, boulevard Preuilly
BP 1701 — 37017 Tours Cedex
☎ 47.37.08.08

Photos
CP & JLB

Dessins
Gilles Coursier

Photogravure
Graphotec – 21, chemin de la Tour
92350 Le Plessis-Robinson
☎ (1) 46 30 44 49

Pom's est une revue indépendante non rattachée à Apple Computer, Inc. ni à Apple Computer France S.A.R.L.
Apple, le logo Apple, Mac et le logo Macintosh sont des marques déposées d'Apple Computer, Inc.
IBM est une marque déposée de International Business Machine.
PC et AT sont des marques déposées de la Société IBM.

©Éditions MEV 1989
Toute reproduction intégrale ou partielle, effectuée par quelque procédé que ce soit, sans l'accord écrit d'Éditions MEV, constitue une contrefaçon.
Loi du 11 mars 1957, articles 425 et suivants du Code Pénal.
Droits de traduction, de reproduction et d'adaptation réservés pour tous pays.

Éditorial

Apple Expo

200 exposants, dont Pom's, attendent 50 000 visiteurs cette année dans le nouveau CNIT de La Défense. Apple n'organise pas moins de 25 conférences sur des thèmes aussi variés que la programmation d'HyperCard, la Musique, les handicapés, Unix, Numéris ou encore l'intelligence artificielle et bien d'autres encore. Ce sera l'occasion d'entendre John Sculley, Michael Spindler d'Apple Europe, Jean-Louis Gassée ou Alan Kay l'un des trois 'penseurs' d'Apple.

Pour ce numéro d'Apple Expo, Pom's met le GS à l'honneur : une application au look Mac, un accessoire de bureau, un programme d'initialisation et un nouveau lot de disquettes freeware et shareware.

Pour tous les Apple //, Pom's a traduit ProCommand de Glen Bredon que vous étiez nombreux à nous demander. Bel ensemble pour un Applesoft toujours jeune. Une commande externe ProDOS par l'une des rares auteuses de la revue et les toujours astucieuses macro-commandes AppleWorks de notre spécialiste Dimitri Geystor.

Côté Mac, un accessoire de bureau indispensable écrit en C, un essai de ConcertWare et des infos, d'autant plus séduisantes qu'Apple annonce une baisse sensible des prix Mac : de – 27 % sur le Macintosh Plus à – 10 % sur le Mac SE 30. À 10 900,00 F le Mac Plus... qui résiste ?

Hervé Thiriez

Bram Master, suite : GS-Init

Jean-Yves Bourdin

Confier votre Tableau de Bord à J.-Y. Bourdin, est-ce bien raisonnable...

Le programme ci-dessous est une extension du programme "Bram Master" publié dans Pom's n° 41 : son but est de mettre en un seul programme les routines d'initialisation ou de réinitialisation de la machine dont je me sers le plus fréquemment.

Je n'ai en effet pas seulement à rétablir ma configuration de Bram : je me suis également aperçu que j'avais fréquemment à "purger" la mémoire du GS. Sans quoi certaines applications, ou bien refusaient de se charger, ou bien plantaient, malgré mes 2,3 Mégas de mémoire en ligne. Il apparaît en effet que certaines applications gèrent de façon pour le moins cavalière (fautive, en fait) les appels au gestionnaire de mémoire du GS, le Memory Manager. Parmi celles-ci, AppleWorks, PaintWorks Gold, et le Linker d'APW...

Au départ, ce programme incluait une routine de configuration du port série pour lui donner un buffer de 64Ko, inspirée de celle de Gary Little. La routine marchait, mais le buffer interne du GS est trop bugué : il vaut mieux ne pas s'en servir du tout.

En revanche, j'ai ajouté quatre sous-routines pour configurer la carte accélérateur TransWarp GS, d'Applied Engineering, de façon que je puisse lancer Thexder sans me faire ridiculiser ensuite.

Obéir au Bureau Politique

Les "bidouilles" avec le Memory Manager et les outils GS ne sont plus de saison. Avec GS/OS, le niveau de maturité du système du GS interdit désormais les fantaisies. Quoi que nous en ayons, il est temps désormais de nous plier, serait-ce en maugréant, aux directives du Bureau Politique. Sans quoi, c'est l'utilisateur qui paye, par des "plantages" intempestifs, l'indiscipline ou l'impatience des programmeurs.

C'est d'ailleurs pourquoi je prêche d'exemple en m'y soumettant moi-même. Le "noyau" de GS.INIT est composé du lancement de trois outils (Tool Locator, Miscellaneous Toolset et Memory Manager), puis de leur fermeture, dans l'ordre spécifié par le Bureau Politique.

«Que d'octets inutiles !», dira le programmeur formé à la chasse aux octets sur Apple //. Effectivement, je n'ai encore jamais vu de situation où cette "bande des trois" ne soit pas déjà active : ils sont toujours là. Mais

Apple nous demande de le faire, faisons-le. Ceci dit, si vous ne le faites pas dans le programme que vous taperez pour vous, c'est votre affaire...

De la même façon, le programme passe son temps à alterner entre mode 16 bits et mode 8 bits (je préfère cette formule à la formule "mode natif/mode émulation" : en mode 8 bits, toutes les instructions du 65816 fonctionnent, il ne se comporte pas comme une "émulation" qui ne reconnaîtrait que les instructions du 6502).

Il fallait choisir un "standard" pour les sous-routines, j'ai choisi le huit bits. Là aussi, que d'octets inutiles : mais que de plantages évités si on fait constamment attention à bien préciser au 65816, et à notre assembleur, en quel mode il doit être.

Un programme GS, il faut s'y faire, est toujours un peu verbeux, et gâche quelques octets. C'est la condition pour s'y repérer, et puis on compte la RAM en Mégas, désormais. Attention, j'ai dit "un peu" verbeux, et "quelques" octets : cela n'excuse pas les abus "à la PaintWorks Gold" !

Purger la mémoire

Précisons, pour ceux qui ne le savent pas, ce qu'est la purge mémoire : une application comme le Finder, par exemple, a besoin de certains outils qui sont sur la disquette système, et il les charge en mémoire. Quand on quitte le Finder, en lançant un programme par exemple, celui-ci devrait dire au système de libérer l'espace mémoire occupé par ces outils, et par le Finder lui-même.

Oui, mais l'application suivante va peut-être avoir besoin des mêmes outils, qu'il va falloir alors recharger à nouveau du disque. De plus, en quittant l'application suivante, on va retourner au Finder, qu'il va falloir encore recharger du disque.

Le Memory Manager permet donc de laisser certains segments de programme, certaines ressources système, chargés en mémoire, mais

VersionSoft, ça plante !

Exemple parfait de ce qu'il ne faut pas faire avec le Memory Manager du GS :

```
00/1FA4 :A9 03 00        LDA #0003    ; charge la valeur 03/0000
00/1FA7 :8F F8 0A 00     STA 000AF8   ; dans une variable
00/1FAB :A9 00 00        LDA #0000
00/1FAE :8F F6 0A 00     STA 000AF6
00/1FB2 :48              PHA          ; démarre appel à NewHandle
00/1FB3 :48              PHA
00/1FB4 :F4 0A 00        PEA 000A     ; longueur du bloc demandé
00/1FB7 :F4 00 00        PEA 0000     ; = 640Ko
00/1FBA :AD 82 C5        LDA C582     ; pour mon identificateur
00/1FBD :48              PHA
00/1FBE :F4 02 80        PEA #%10000000_00000010 ; attributs :
                                       verrouillé, adresse fixe
00/1FC1 :AF F8 0A 00     LDA 000AF8   ; à créer à partir de
                                       l'adresse fixe
00/1FC5 :48              PHA          ; 03/0000 (début du banc 3)
00/1FC6 :AF F6 0A 00     LDA 000AF6
00/1FCA :48              PHA
00/1FCB :A2 02 09        LDX #0902    ; appel à
00/1FCE :22 00 00 E1     JSL $E10000  ; NewHandle
00/1FD2 :90 12           BCC 1FE6     ; si ok, passe à la suite
00/1FD4 :68              PLA
00/1FD5 :68              PLA
00/1FD6 :AF F8 0A 00     LDA 000AF8   ; sinon incrémente partie
00/1FDA :1A              INC          ; haute de l'adresse du banc
                                        mémoire demandé
00/1FDB :8F F8 0A 00     STA 000AF8   ; pour le début de mon bloc
00/1FDF :C9 08 00        CMP #0008    ; mais on ne peut pas
                                        occuper au-dessus
00/1FE2 :D0 CE           BNE 1FB2     ; de $03 + $0A + $ 07 = $15.
00/1FE4 :00 4C           BRK 4C       ; Sinon, BREAK : tu plantes
                                        la machine ! ! !
00/1FE6 :68              PLA          ; il doit rester de la place
                                        pour charger
00/1FE7 :8F FA 0A 00     STA 000AFA   ; les outils et le système.
00/1FEB :68              PLA
00/1FEC :8F FC 0A 00     STA 000AFC
00/1FF0 :F4 00 00        PEA 0000
00/1FF3 :F4 A6 0A        PEA 0AA6
00/1FF6 :A2 01 0E        LDX #0E01
00/1FF9 :22 00 00 E1     JSL E10000   ; LoadTools
```

Ce qui se trouve ci-dessus est le désassemblage commenté de ce que j'ai pu trouver, suite à une remarque d'Yvan Koenig, à l'octet $1FA4 dans ma version de PaintWorks Gold. Une conjonction exemplaire de deux énormes erreurs volontaires :

1° Faire comme si il ne pouvait pas y avoir d'extension mémoire supérieure à 1,2 Mégas pour le GS (!)

2° En cas d'insuffisance de mémoire libre en-dessous de ce 1,2 Méga, ne pas quitter proprement, ne rien dire à l'utilisateur, même pas renvoyer à une erreur ProDOS : planter, tout simplement. Ça plante : c'est étudié pour !

Et je ne parlerai pas de la pratique qui consiste à demander un bloc de 640Ko, d'un seul tenant, commençant à une adresse fixe, au Memory Manager, en lui interdisant même de déplacer ce bloc en mémoire ensuite... Voir "Le mange-mémoire" dans Pom's 37 page 59.

Vous avez donc bougrement intérêt, si vous utilisez des logiciels VersionSoft, à les lancer par GS.INIT, avec la routine de purge et de compactage **préalables** de la mémoire. Mais PaintWorks Gold est un excellent logiciel par ailleurs. Vous trouverez un patch pour corriger son erreur dans Pom's 42 page 64.

dans un état "purgeable" : si l'application suivante n'a pas besoin de toute la mémoire disponible, elle trouvera ses outils déjà chargés en mémoire. Et l'on pourra retourner quasi-instantanément au Finder en sortant.

Si en revanche elle a besoin de toute la mémoire, le Memory Manager "purgera" ces segments purgeables et rangera ("compactera") la mémoire disponible pour faire de la place. Le Memory Manager raffine même, en fixant 3 niveaux différents de purge, ce qui permet de fixer des priorités dans la purge.

Toutes ces subtilités permettent d'aller le plus vite possible avec un espace mémoire restreint. Elles signifient que plus vous aurez de mémoire, plus le GS tournera vite, et moins il fera d'accès disque. Ce sont ces subtilités qu'ignorent les applications "primaires", comme AppleWorks ou PaintWorks Gold, dont le seul souci est de manger pour elles-mêmes le plus de mémoire possible. Il faut donc purger nous-mêmes, avant et/ou après elles.

Comment purger ?

J'utilisais auparavant la routine de purge mémoire de Glen Bredon dans Merlin 16 Plus et de Mike Westerfield dans Call Apple d'Avril 89, mais j'y ai renoncé. Cette routine purge la mémoire bloc par bloc, en cherchant pour chacun des "handles" concernés s'il est purgeable ou non.

Cette routine marche, et elle a l'avantage de permettre si on le veut d'afficher à chaque fois quel bloc on libère. Mais elle utilise la connaissance des enregistrements du Memory Manager : or cette structure n'est absolument pas garantie par Apple, et il est possible que cette routine ne marche plus dans les versions ultérieures des outils. Toujours le soumission au Bureau Politique...

Bientôt d'ailleurs, je l'espère, nous n'aurons plus à purger la mémoire : les dernières versions du Memory Manager ont en effet un appel "RealMem", distinct du classique "FreeMem", qui renvoie non seulement l'espace mémoire libre, mais l'espace mémoire libérable au total après purge et compactage. Une application pourra donc savoir exactement sur combien de mémoire elle peut compter.

J'ai donc choisi pour purger la mémoire une formule qui devrait rester compatible, tout en restant évidemment dans une certaine mesure une bidouille (nous sommes sur Apple //, n'est-ce pas).

Au lieu que le programme purge lui-même, bloc après bloc, mon idée a été de demander au Memory Manager de m'allouer un bloc mémoire de... 16 Mégas ! Bien entendu, le Memory

GS.INIT

```
                                 ***************************************************
                                 *                                                 *
                                 * GS INIT.                                        *
                                 *                                                 *
                                 * Ce programme se charge de mettre ou remettre le *
                                 * GS dans la configuration choisie par            *
                                 * l'utilisateur.                                  *
                                 *                                                 *
                                 * Il comprend un noyau standard qui lance et ferme*
                                 * les outils, et un choix des différentes formes  *
                                 * de Quit possibles sous Prodos 8. Il incorpore   *
                                 * pour le moment trois sous-routines :            *
                                 *                                                 *
                                 * - PURGE.MEM : Purge et compacte la mémoire      *
                                 * - MODBRAM : Reconfiguration automatique de la   *
                                 *             mémoire sauvegardée                 *
                                 * - Trois sous-routines de configuration de la    *
                                 *             carte Transwarp GS Appplied.        *
                                 *                                                 *
                                 * Assembleur Merlin 16 Plus.                      *
                                 * J.Y. Bourdin, Pom's 44.                         *
                                 *                                                 *
                                 ***************************************************


                =BF00           MLI      EQU   $BF00
                =E10000         TOOLBOX  EQU   $E10000

                                         MX    %11        ; on démarre en mode
                                                          ; émulation (8 bits)
                                         ORG   $2000


                                         DSK   GS.INIT
                                         TYP   $FF

002000 : 20 58 FC                        JSR   $FC58      ; Home

002003 : 38                     :VERIFGS SEC              ; vérifie qu'on est sur
002004 : 20 1F FE                        JSR   $FE1F      ; GS
002007 : 90 03 =200C                     BCC   INTRO
002009 : 4C 64 20                        JMP   P8QUIT     ; sinon quitte

00200C : 18                     INTRO    CLC              ; passage en 16 bits
00200D : FB                              XCE
00200E : C2 30                           REP   %00110000
002010 : A2 01 02                        LDX   #$0201     ; TLStartup : démarre le
002013 : 22 00 00 E1                     JSL   TOOLBOX    ; Tool Locator
002017 : F4 00 00                        PEA   #$0000
00201A : A2 02 02                        LDX   #$0202     ; MMStartUp : démarre le
00201D : 22 00 00 E1                     JSL   TOOLBOX    ; Memory Manager
002021 : 68                              PLA
002022 : 8D 62 20                        STA   MYID
002025 : A2 03 02                        LDX   #$0203     ; MTStartup : démarre le
002028 : 22 00 00 E1                     JSL   TOOLBOX    ; Misc Toolset

00202C : 38                              SEC              ; repasse en 8 bits, car
00202D : FB                              XCE              ; les sous-routines
                                                          ; démarrent en 8 bits
```

Manager n'est pas près de pouvoir me le donner. Mais, avant de me le refuser, il commence par tout "tasser" (purger et compacter) de lui-même. C'est seulement ensuite qu'il s'aperçoit qu'il n'a pas assez de mémoire : mais la purge est faite.

Au cas où le Memory Manager, un jour, aurait 16 Mégas à m'allouer (!), le programme les lui redonne aussitôt : dans ce cas donc, tout ne serait pas purgé, mais on aurait 16 Mégas de libres, ce qui devrait suffire pour nos applications...

L'inconvénient de ce choix, c'est que l'utilisateur de la routine ne voit rien du tout, il n'y a pas d'affichage des blocs libérés. Mais c'est précisément ce dont j'avais besoin : on purge et on quitte, c'est tout, pas besoin de raconter sa vie.

Un bug d'AppleWorks

Quelqu'un (peut-être votre serviteur ?) a signalé un jour que quand on lançait AppleWorks deux fois de suite sous ProDOS 16, on gagnait environ 100Ko pour le bureau. AppleWorks avait apparemment trouvé un truc pour augmenter "RealFreeMem" (qui s'accroît de 100Ko après le passage d'AppleWorks). C'est vrai, mais ensuite... le plantage guette.

AppleWorks fait en effet une bêtise "de débutant" dans la gestion de la mémoire du GS : quand une application démarre le Memory Manager (voir les lignes 50 à 54 ci-dessous), celui-ci répond en lui accordant un numéro d'identification (UserId). Et AppleWorks a parfaitement raison, en quittant, de dire au Memory Manager qu'il peut disposer de la mémoire qu'il avait accordée à l'application (DisposeAll), avant de clore le Memory Manager.

Oui, mais AppleWorks est une application ProDOS 8, et non ProDOS 16. Si on a démarré sous GS/OS par exemple, le Memory Manager attribue systématiquement aux applications ProDOS 8 qui font un MMStartup l'identificateur de ProDOS lui-même ($3001).

```
00202E : 20 EC 20      63            JSR   PURGEMEM
                       64
002031 : 20 35 21      65            JSR   MODBRAM
                       66
002034 : 20 75 22      67            JSR   TWFAST
                       68
002037 : 20 8D 22      69            JSR   TWSLOW
                       70
00203A : 20 A8 22      71            JSR   IRQFAST
                       72
00203D : 20 BC 22      73            JSR   IRQSLOW
                       74
                       75
002040 : 18            76   FIN      CLC                ; passage en 16 bits
002041 : FB            77            XCE
002042 : C2 30         78            REP   %00110000
002044 : A2 03 03      79            LDX   #$0303       ; MTShutDown
002047 : 22 00 00 E1   80            JSL   TOOLBOX
00204B : AD 62 20      81            LDA   MYID
00204E : 48            82            PHA
00204F : A2 02 03      83            LDX   #$302        ; MMShutDown
002052 : 22 00 00 E1   84            JSL   TOOLBOX
002056 : A2 01 03      85            LDX   #$0301       ; TLShutDown
002059 : 22 00 00 E1   86            JSL   TOOLBOX
00205D : 38            87            SEC                ; retour en 8 bits
00205E : FB            88            XCE
00205F : 4C 74 20      89            JMP   LAUNCH       ; ou P8QUIT ou HRDRBT
                       90                               ; ou APLRBT
002062 : 00 00         91   MYID     DS    2,#$00
                       92
                       93   ************************************************
                       94
                       95   * Choix du Quit
                       96
                       97            MX    %11          ; assemblage 8 bits
                       98
002064 : 20 00 BF      99   P8QUIT   JSR   MLI          ; Quit Prodos 8 ordinaire
002067 : 65            100           DFB   #$65
002068 : 6D 20         101           DA    :PARM
00206A : B0 00 =206C   102           BCS   :ERR
00206C : 00            103  :ERR     BRK
                       104
00206D : 04            105  :PARM    DFB   #$04
00206E : 00            106           DFB   #$00
00206F : 00 00         107           DA    $0000
002071 : 00            108           DFB   #$00
002072 : 00 00         109           DA    $0000
                       110
002074 : 20 00 BF      111  LAUNCH   JSR   MLI          ; Quit Prodos 8
002077 : 65            112           DFB   #$65         ; "Enhanced"
002078 : 7D 20         113           DA    :PARM
00207A : B0 00 =207C   114           BCS   :ERR
00207C : 00            115  :ERR     BRK
00207D : 04            116  :PARM    DFB   #$04
00207E : EE            117           DFB   #$EE
00207F : 84 20         118           DA    :NAME
002081 : 00            119           DFB   #$00
002082 : 00 00         120           DA    $0000
                       121
002084 : 0F 2F 53 2F   122  :NAME    STR   '/S/SYSTEM/START' ;mettre le chemin
002088 : 53 59 53 54 45 4D 2F 53
002090 : 54 41 52 54
                       123                              ; d'accès complet du
                       124                              ; fichier à lancer
002094 : EA EA EA EA   125           DS    30,#$EA
002098 : EA EA EA EA EA EA EA EA
0020A0 : EA EA EA EA EA EA EA EA
0020A8 : EA EA EA EA EA EA EA EA
0020B0 : EA EA
                       126
0020B2 : 18            127  HRDRBT   CLC                ; passage en 16 bits
0020B3 : FB            128           XCE
0020B4 : C2 30         129           REP   %00110000
                       130
0020B6 : A9 51 00      131           LDA   #$0051       ; Hard Reboot de Bill
```

Quand une application dit au Memory Manager de libérer la mémoire occupée par ProDOS, il nettoie tout l'espace mémoire occupé par le système, y compris l'espace mémoire en page zéro occupé par ProDOS 8 lui-même. Le Memory Manager pense par exemple que la page 00/BF00 est libre, et l'attribue au deuxième lancement d'AppleWorks, comme tout le reste de la mémoire, au bureau d'AppleWorks ! Faites une carte-mémoire du GS sous AppleWorks lancé pour la seconde fois, et vous aurez des surprises... On gagne 100Ko, mais à quel prix !

Une application Système 16 n'a pas ce problème, son identificateur n'étant pas celui de ProDOS. Mais, dans tous les cas, le mieux est bien de suivre la procédure recommandée par le Bureau Politique (Toolbox Reference, vol 1) : avant toute demande de mémoire par un appel NewHandle, commencer par modifier le champ auxiliaire du numéro d'identification, ou demander un nouvel identificateur par l'intermédiaire du Miscellaneous Toolset (GetNewId).

Au fait, oui, un mot de la solution : il faudrait sans doute patcher AppleWorks, plus exactement réécrire toute la partie de gestion de la mémoire du GS, pour la mettre en conformité avec les règles Apple. Je crois que cela a été fait avec la version 3.0.

Pour ma part, je m'en tire en lançant AppleWorks depuis un fichier S16 qui s'installe en mémoire, et auquel AppleWorks renvoie donc obligatoirement en sortie. Etant un fichier S16, il relance le système dès qu'AppleWorks quitte. Si vous lancez AppleWorks depuis le Finder ou depuis Prosel 16, ou tout autre lanceur S16, ça ira aussi. Mais ça n'ira pas si vous le lancez directement depuis Prosel 8.

```
0020B9 : 85 00          132          STA   $00          ; Basham pas du tout
0020BB : F4 02 00       133          PEA   $0002        ; garanti par Apple
0020BE : F4 00 00       134          PEA   $0000
0020C1 : F4 00 00       135          PEA   $0000
0020C4 : F4 08 00       136          PEA   $0008
0020C7 : A2 09 09       137          LDX   #$0909
0020CA : 22 00 00 E1    138          JSL   TOOLBOX
                        139
0020CE : 38             140          SEC               ; retour en 8 bits
0020CF : FB             141          XCE
                        142
0020D0 : 78             143  APLRBT  SEI               ; Reboot recommandé par
0020D1 : A9 0C          144          LDA   #$0C        ; Apple sur GS
0020D3 : 8D 68 C0       145          STA   $C068       ; StateReg
0020D6 : 9C 47 C0       146          STZ   $C047       ; ClrVblInt
0020D9 : 9C 41 C0       147          STZ   $C041       ; IntEn
0020DC : A9 09          148          LDA   #$09
0020DE : 8D 39 C0       149          STA   $C039       ; SccaReg
0020E1 : A9 C0          150          LDA   #$C0
0020E3 : 8D 39 C0       151          STA   $C039
0020E6 : CE F4 03       152          DEC   $03F4
0020E9 : 6C FC FF       153          JMP   ($FFFC)
                        154
                        155  ************************************************
                        156  *                                              *
                        157  * ROUTINE PURGE.MEM                           *
                        158  * Force le Memory Manager à purger les blocs   *
                        159  * purgeables et à compacter la mémoire         *
                        160  * en lui demandant un bloc mémoire supérieur   *
                        161  * à ce qu'il peut attribuer.                   *
                        162  *                                              *
                        163  ************************************************
                        164
                        165          MX    %11         ; sous-routines démarrent
                        166                            ; en 8 bits
0020EC : 18             167  PURGEMEM CLC              ; passe en 16 bits
0020ED : FB             168          XCE
0020EE : C2 30          169          REP   %00110000
                        170
                        171  * demande un nouvel identificateur
                        172
0020F0 : F4 00 00       173          PEA   #$0000
0020F3 : F4 00 10       174          PEA   #%00010000_00000000 ; idTag : applic
0020F6 : A2 03 20       175          LDX   #$2003      ; GetNewId
0020F9 : 22 00 00 E1    176          JSL   TOOLBOX
0020FD : 68             177          PLA               ; range l'identificateur
0020FE : 8D 33 21       178          STA   :MYID
                        179
                        180  * demande un bloc de 16 Mégas
                        181
002101 : F4 00 00       182          PEA   #$0000
002104 : F4 00 00       183          PEA   #$0000
002107 : F4 FF 00       184          PEA   #$00FF      ; taille du bloc :
00210A : F4 FF FF       185          PEA   #$FFFF      ; 16 Mégas !
00210D : AD 33 21       186          LDA   :MYID       ; notre identificateur
002110 : 48             187          PHA
002111 : F4 08 13       188          PEA   #%00010011_00001000 ; application
                        189                            ; niveau de purge 3
002114 : F4 00 00       190          PEA   #$0000      ; LocationPtr : sans
002117 : F4 00 00       191          PEA   #$0000      ; importance ici
00211A : A2 02 09       192          LDX   #$0902      ; Newhandle
00211D : 22 00 00 E1    193          JSL   TOOLBOX
002121 : B0 00 =2123    194          BCS   :ERROR      ; erreur : toujours
                        195
002123 : 68             196  :ERROR  PLA               ; laisser la pile propre
002124 : 68             197          PLA
002125 : AD 33 21       198          LDA   :MYID       ; désalloue la mémoire
002128 : 48             199          PHA               ; de cet identificateur
002129 : A2 02 11       200          LDX   #$1102      ; DisposeAll
00212C : 22 00 00 E1    201          JSL   TOOLBOX
                        202
002130 : 38             203  :END    SEC               ; retour en 8 bits
002131 : FB             204          XCE
002132 : 60             205          RTS
                        206
002133 : 00 00          207  :MYID   DS    2,#$00
                        208
                        209
                        210  ************************************************
                        211  *                                              *
                        212  * RECONFIGURATION DE LA BRAM.                  *
                        213  * Cette routine reconfigure la Ram Batterie    *
```

À propos de la Transwarp

Les sous-routines utilisées pour configurer la carte sont celles trouvées dans la documentation.

J'ai du même coup trouvé un bug sérieux du "firmware" de ma carte : la sous-routine "ResetTW", qui est censée remettre la TransWarp dans la configuration indiquée par l'utilisateur dans le tableau de bord de la TransWarp, plante purement et simplement.

Pour voir si votre Transwarp a le même bug, passez sous Basic System et tapez :

```
$CALL-151
*<CTRL-N> (pour avoir des registres
    16 bits)
*BC/FF0CX
```

En revanche, je m'étais laissé dire que la routine d'identification de la Transwarp proposée par AE était fausse : elle marche impeccablement. À mon avis, certains doivent utiliser un assembleur qui n'est pas le bon...

Vous trouverez les explications concernant l'option dite "Appletalk/ IRQ" de la Transwarp dans Pom's 43 pages 62-63.

Présentation du listing

La partie concernant les différents Quit en sortie n'a pas changé depuis Pom's 41, prière donc de se reporter à ce numéro pour toutes explications.

En revanche la partie "ModBram" qui modifie les réglages du tableau de bord, a subi quelques modifications du fait de son passage en sous-routine. Mais les explications de Pom's 41 suffisent.

Le "noyau" central du programme est conçu de telle façon qu'on puisse lui rajouter d'autres sous-routines en fonction des besoins, ou supprimer ce qui n'est pas souhaité. Il suffit de supprimer les JSR correspondants, ou de les transformer en BIT.

On peut patcher le code objet avec des $2C à la place des $20 en $202E et suivants. Pour un programme à

```
                          * sans passer par le tableau de bord du GS,        *
                          * et fait prendre en compte la modification par     *
                          * le GS. Pour que Prodos 16 et 8 reconfigurent      *
                          * leur liste de "devices" aussitôt, lancer          *
                          * ensuite un programme de type "Start" de P16       *
                          * (voir l'étiquette Launch). Cf Pom's 41.           *
                          *                                                   *
                          ***************************************************

                                   MX     %11          ; sous-routines démarrent
                                                       ; en mode 8 bits
002135 : 18               MODBRAM  CLC                 ; passage en 16 bits
002136 : FB                        XCE
002137 : C2 30                     REP    %00110000

002139 : F4 00 00         :READBRM PEA    $0000
00213C : F4 75 21                  PEA    :BRAMBUF
00213F : A2 03 0A                  LDX    #$0A03       ; lit la Bram
002142 : 22 00 00 E1               JSL    TOOLBOX

002146 : E2 30            :CHGBRM  SEP    %00110000    ; registres 8 bits
002148 : A9 00                     LDA    #$00         ; pas de buffer sur le
00214A : 8D 7A 21                  STA    :BRAMBUF+$05 ; port 1
00214D : A9 00                     LDA    #$00         ; met l'affichage en
00214F : 8D 8F 21                  STA    :BRAMBUF+$1A ; "écran Mac"
002152 : A9 0F                     LDA    #$0F
002154 : 8D 90 21                  STA    :BRAMBUF+$1B
002157 : A9 0F                     LDA    #$0F
002159 : 8D 91 21                  STA    :BRAMBUF+$1C

00215C : C2 30            :WRTBRM  REP    %00110000    ; registres 16 bits
00215E : F4 00 00                  PEA    $0000
002161 : F4 75 21                  PEA    :BRAMBUF
002164 : A2 03 09                  LDX    #$0903       ; écrit la Bram
002167 : 22 00 00 E1               JSL    TOOLBOX

00216B : E2 30            :SETBRM  SEP    %00110000    ; registres 8 bits
00216D : 18                        CLC                 ; avec CLC, reconfigure
                                                       ; les slots
00216E : 22 94 00 E1               JSL    $E10094      ; vecteur ToBramSetup

002172 : 38               :END     SEC                 ; repasse en 8 bits
002173 : FB                        XCE
002174 : 60                        RTS

002175 : EA EA EA EA      :BRAMBUF DS     256,#$EA
    (.......)
002271 : EA EA EA EA

                          ***************************************************
                          *                                                 *
                          * TRANSWARP FAST : cette routine reconfigure la   *
                          * carte accélérateur Transwarp GS d'Applied       *
                          * Engineering pour tourner à sa vitesse maximum   *
                          *                                                 *
                          ***************************************************

                                   MX     %11          ; Assemblage 8 bits

002275 : 18               TWFAST   CLC                 ; passage en 16 bits
002276 : FB                        XCE
002277 : C2 30                     REP    %00110000

002279 : AF 00 FF BC               LDA    >$BCFF00     ; vérifie la présence de
00227D : C9 54 57                  CMP    #'TW'        ; la carte Transwarp
002280 : D0 08 =228A               BNE    :END

002282 : 22 10 FF BC               JSL    $BCFF10      ; GetMaxSpeed
002286 : 22 24 FF BC               JSL    $BCFF24      ; SetCurSpeed

00228A : 38               :END     SEC                 ; repasse en 8 bits
00228B : FB                        XCE
00228C : 60                        RTS
```

publier dans une revue, cette formule me semble préférable à celle des "DO" à l'assemblage.

C'est volontairement que ce programme-source n'inclut pas de macros : j'aime bien savoir ce que je fais, et cela le rend plus lisible.

Utilisateurs, attention : ce que fait le programme est parfaitement invisible, tout se passe "dans la tête" de la machine. Seul le passage (à supprimer) en "écran Mac" donne un indice qu'il se passe quelque chose. Ce programme n'est pas à utiliser tel quel, mais à ajuster en fonction de votre configuration et de vos besoins. Et si vous y rajoutez d'autres sous-routines, envoyez-les nous...

Le programme-objet correspondant au listing ci-contre... n'existe pas ! Il serait absurde, puisqu'il consisterait à défaire ce qu'il vient de faire. À la place, sur la disquette Pom's, vous trouverez six fichiers correspondant aux six fonctions du programme, (*PURGE.MEM*, *MODBRAM*, *TRANSWARP FAST*, *TRANSWARP SLOW*, *TW.IRQ.FAST*, *TW.IRQ.SLOW*).

Références

- Sur la partie "ModBram" et les divers Quit, voir Pom's 41.
- Sur le Memory Manager et le Miscellaneous ToolSet, voir Toolbox Reference, volume 1, et la mise à jour Apple IIG S Toolbox Reference Update (pour RealFreeMem par exemple ; un indispensable).

NB : pour avoir une synthèse complète et rapide des appels outils, plutôt que les deux volumes Apple de Quick Reference (C et Assembleur), je conseille le Programmer's OnLine Companion pour le C et le Pascal, Merlin 16 et ses ToolHelp (qu'on peut adapter pour APW) pour l'assembleur.

- Sur la carte Transwarp GS : voir le "Programmer's Reference", inclus dans la documentation française.

```
                                 ************************************************
                                 *                                              *
                                 * TRANSWARP SLOW : cette routine reconfigure la *
                                 * carte accélérateur Transwarp GS d'Applied    *
                                 * Engineering pour tourner à 2,6 MHz (vitesse  *
                                 * "rapide" du GS).                             *
                                 *                                              *
                                 ************************************************

                                         MX    %11          ; Assemblage 8 bits

00228D : 18                      TWSLOW  CLC               ; passage en 16 bits
00228E : FB                              XCE
00228F : C2 30                           REP   %00110000

002291 : AF 00 FF BC                     LDA   >$BCFF00     ; vérifie la présence de
002295 : C9 54 57                        CMP   #'TW'        ; la carte Transwarp
002298 : D0 0B =22A5                     BNE   :END

00229A : A2 01 00                        LDX   #1           ; vitesse 2,6 MHz
00229D : 22 1C FF BC                     JSL   $BCFF1C      ; Index2Freq
0022A1 : 22 24 FF BC                     JSL   $BCFF24      ; SetCurSpeed

0022A5 : 38                      :END    SEC               ; repasse en 8 bits
0022A6 : FB                              XCE
0022A7 : 60                              RTS

                                 ************************************************
                                 *                                              *
                                 * IRQFAST : cette routine reconfigure la       *
                                 * carte accélérateur Transwarp GS d'Applied    *
                                 * Engineering pour ne pas se ralentir durant les *
                                 * interruptions.                               *
                                 *                                              *
                                 ************************************************

                                         MX    %11          ; Assemblage 8 bits

0022A8 : 18                      IRQFAST CLC               ; passage en 16 bits
0022A9 : FB                              XCE
0022AA : C2 30                           REP   %00110000

0022AC : AF 00 FF BC                     LDA   >$BCFF00     ; vérifie la présence de
0022B0 : C9 54 57                        CMP   #'TW'        ; la carte Transwarp
0022B3 : D0 04 =22B9                     BNE   :END

0022B5 : 22 34 FF BC                     JSL   $BCFF34      ; supprime le ralentisse-
                                                            ; ment de la Transwarp
                                                            ;durant les interruptions

0022B9 : 38                      :END    SEC               ; repasse en 8 bits
0022BA : FB                              XCE
0022BB : 60                              RTS

                                 ************************************************
                                 *                                              *
                                 * IRQSLOW : cette routine reconfigure la       *
                                 * carte accélérateur Transwarp GS d'Applied    *
                                 * Engineering pour se ralentir durant les      *
                                 * interruptions.                               *
                                 *                                              *
                                 ************************************************

                                         MX    %11          ; Assemblage 8 bits

0022BC : 18                      IRQSLOW CLC               ; passage en 16 bits
0022BD : FB                              XCE
0022BE : C2 30                           REP   %00110000

0022C0 : AF 00 FF BC                     LDA   >$BCFF00     ; vérifie la présence de
0022C4 : C9 54 57                        CMP   #'TW'        ; la carte Transwarp
0022C7 : D0 04 =22CD                     BNE   :END

0022C9 : 22 38 FF BC                     JSL   $BCFF38      ; rétablit le ralentisse-
                                                            ; ment de la Transwarp
                                                            ;durant les interruptions

0022CD : 38                      :END    SEC               ; repasse en 8 bits
0022CE : FB                              XCE
0022CF : 60                              RTS

                                 ************************************************
```

Les nouveaux TimeOut

Dimitri Geystor

Comme le dit la sagesse populaire, un bonheur n'arrive jamais seul...

Pom's vous propose un joli doublé pour la rentrée 1989 : **MacroTools.II** et **DeskTools.II** dans leur version française, qui viennent compléter une panoplie déjà impressionnante d'applications TimeOut. Deux disquettes truffées de morceaux de choix. En voici un survol, à l'intention de tous les adeptes d'AppleWorks.

MacroTools.II

Une vingtaine de macros fort diverses feront la joie des programmeurs, mais aussi de tous ceux qui se préoccupent moins du "comment ça marche" et souhaitent plus prosaïquement s'équiper d'outils "prêts à l'emploi".

Pour les premiers (par prédilection), deux macros tout à fait inédites :

- une macro "Désassembleur", qui traduit automatiquement n'importe quelle petite routine écrite en langage machine en "pokes" compréhensibles pour UltraMacros ;
- une macro "ClavierInterdit", qui détourne la fonction habituelle des macros <ato> pour en faire des sous-routines inaccessibles à partir du clavier et uniquement "appelables" par une autre macro, d'où un double avantage : aucun risque de déclencher par inadvertance une sous-routine malencontreuse, et disponibilité d'un jeu complet de définitions pour les sous-routines (qui ne feront jamais double emploi avec les macros à "part entière").

Pour les deuxièmes (mais non exclusivement, bien sûr), une macro d'impression de Traitement de textes sur deux colonnes, une macro de chaînage de Bases de données "en parallèle", histoire de disposer de "pseudo-fiches" de 60, voire de 90 catégories...

Pour tout le monde, sans ostracisme, des macros pour caler la date et l'heure d'AppleWorks, avec et sans horloge. Et pour les possesseurs de GS, la saisie de l'heure à la seconde près.

Des macros encore pour :

- copier l'écran dans le presse-papiers ;
- envoyer des codes à votre imprimante ;
- utiliser l'imprimante comme une vulgaire machine à écrire ;
- charger d'un coup toutes vos applications TimeOut même s'il y en a plus de 30, trouver sans douleur et précisément les coordonnées de tout point de l'écran.

Parmi les fichiers de la disquette, vous trouverez aussi un tableau récapitulatif de toutes les macros UltraMacros déjà publiées dans Pom's.

Et –je réservais la bonne surprise pour le dessert– il n'y a pas que des macros sur cette disquette. Vous y trouverez aussi quatre mini-applications TimeOut inédites :

① Avec "Changer BIP" vous pourrez modifier rapidement et précisément le BIP d'AppleWorks.

② Grâce à "Lister.Type" vous saurez lister sélectivement, avec AppleWorks, n'importe quel type de fichier (au lieu de vous heurter à la dénomination "autres" dont il affuble globalement tout ce qui n'est pas TdT, BdD ou Tableur).

③ La mini-application "Références" vous donnera instantanément une "photographie" du statut actuel d'AppleWorks : lecteur actuel, nom d'accès ProDOS actuel, fichier actuel, nombre de fichiers sur le Bureau, occupation et type du Presse-papiers.

④ L'application "Noms d'accès" est modeste par la place-mémoire qu'elle occupe, mais elle m'est vite devenue tout à fait indispensable : elle se présente sous la forme d'un écran, appelable par ⌘-Escape, où vous pouvez inclure à l'avance (et en permanence par configuration, si vous le souhaitez), jusqu'à 12 noms d'accès ProDOS prédéterminés. Pour changer le "chemin actuel" d'AppleWorks, pointez et cliquez : imbattable en simplicité et en rapidité.

DeskTools.II

Il faut le dire tout de suite, le titre anglais "Accessoires de bureau" me semble pêcher par excès de modestie. La calculatrice scientifique/financière "Calculator+" est un authentique programme à part entière (même s'il est appelable à tout moment directement à partir d'AppleWorks, avec lequel il sait échanger ses données). Quant aux autres applications, il serait plus exact de les traiter de prolongements naturels d'AppleWorks, plutôt que d'accessoires, vu la connotation de futilité qui accompagne généralement ce dernier vocable.

Mais jugez plutôt par vous-mêmes :

Accès direct

Vous avez des disques à forte capacité (3,5 pouces, ou disque dur), et malgré tout le soin que vous apportez à vos classements vous ne retrouvez pas toujours facilement tel ou tel fichier ? Vos soucis appartiennent désormais au passé : "Accès direct" se charge de tout.

① "Accès direct" vous demande d'abord quel lecteur, ou quel catalogue vous voulez examiner.

② Il demande ensuite si vous voulez voir :
- les fichiers AppleWorks seuls ;
- tous les fichiers ;
- les fichiers dont le nom contient une chaîne de caractères déterminée.

Ces informations enregistrées, "Accès direct" fait son travail : il lit tout le catalogue avec tous ses sous-catalogues et affiche un superbe tableau de tous les fichiers désignés. Les fichiers des sous-catalogues sont indentés, et précédés d'un, de deux ou de plusieurs points correspondant aux "niveaux" de leurs sous-catalogues.

Ce super-catalogue est imprimable, en entier ou par écrans. On peut aussi le faire défiler à l'écran avec la souris, vers le haut ou vers le bas, autant de fois qu'on le désire.

De plus, les fichiers spécifiquement AppleWorks (TdT, BdD ou Tableur) sont signalés en marge par une petite pomme. Il suffit de cliquer dessus, et aussitôt le fichier sélectionné est chargé sur le Bureau, tandis que son préfixe devient le "chemin actuel" d'AppleWorks !

Je n'en dis pas plus, c'est tout bonnement magique.

Calculator+

Si vous voulez faire vos comptes d'épicerie sur cette calculette, ce n'est certes pas interdit, mais là n'est pas sa véritable vocation.

Il s'agit d'une calculette scientifique et financière à Notation Polonaise Inverse (NPI). Les opérations et les formules les plus complexes s'écrivent au moyen d'une petite pile de 4 variables. On met des données dans les variables, on leur applique une opération : le résultat passe au sommet de la pile tandis que les variables (dont on n'a plus besoin) sont purgées. Et ainsi de suite. Cela ne vous rappelle rien ? Ce n'est pas autrement que procédaient nos grands-parents sur leurs bouliers.

Sauf qu'ici, nous travaillons avec tout le confort de l'électronique, que nous pouvons choisir la base (décimale, hexadécimale), le format (jusqu'à 16 chiffres significatifs, nombre de chiffres après la virgule, notation scientifique), les fonctions (trigonométriques, exponentielle, logarithmique, etc.), les options financières (amortissement, versements, dépôt périodique ou global, valeur actuelle ou acquise).

Les valeurs peuvent être introduites à la main, prélevées dans une cellule de Tableur, et "exportées" dans toute application AppleWorks. Les connaisseurs apprécieront.

Chronomètres

Cette application donne le meilleur d'elle-même sur le IIGS, car elle chronomètre à la seconde près. Elle fonctionnera aussi sur tout autre Apple // muni d'une horloge, avec un précision d'une minute. Vous n'avez pas d'horloge ? Désolé, attendez-moi un instant, on se retrouve à l'application suivante.

① Vous disposez non pas d'un mais de cinq chronomètres, et pouvez mesurer jusqu'à cinq événements distincts. Chaque chrono donne l'heure du début, de la fin et le temps écoulé.

② Vous pouvez même ajouter à chaque événement, une note explicative. Une fois les chronos démarrés, vous pouvez revenir à AppleWorks, et continuer ce que vous aviez à faire : le chronométrage continue. Vous pouvez, bien entendu, consulter à nouveau le chronomètre - ou l'arrêter à votre gré : il suffit de revenir à l'application par ♂-Escape.

Éteindre écran

L'écran s'obscurcit après un compte à rebours de quelques secondes. Pour le réactiver, appuyer sur une touche.

Ah oui, il y a un petit "extra" : on peut configurer cette application pour que l'écran ne revienne que si vous tapez un code secret de votre choix. Tant pis pour les petits

curieux qui voudraient ausculter le travail en cours pendant que vous vous êtes momentanément absenté.

Gérer imprimantes

Avec cette application, vous pouvez envoyer tous les codes que vous voulez directement aux imprimantes définies dans AppleWorks.

La configuration est très riche en possibilités. Vous décrivez les commandes (14 par imprimante), et leur attribuez les codes à envoyer (jusqu'à 10 octets par commande).

À l'appel de l'application, il suffit ensuite de donner le numéro de la commande pour que celle-ci soit immédiatement exécutée, c'est-à-dire transmise à l'imprimante en ligne.

Ceci permet l'envoi à volonté de tous les codes simples (caractères gras, souligné, caractères par pouce, appel du jeu de caractères spéciaux, etc.), mais aussi une gestion complexe du travail de l'imprimante si vous en connaissez un peu les principes.

La configuration est très simple à faire, et je me suis aperçu que cette application est un outil idéal pour tous les "travaux pratiques" indispensables quand on veut perfectionner sa propre compréhension du manuel d'une imprimante.

Imprimer écran

Une application toute simple mais extrêmement utile. En fait, elle permet d'imprimer (avec l'imprimante ⌘-H) n'importe quelle partie de l'écran. On définit un bloc de texte avec les flèches (ou la souris), en le mettant en inverse vidéo. On valide : le bloc sélectionné est immédiatement imprimé.

D'un emploi ultra-facile et intuitif.

Prospecter fichier

Voilà un outil très incisif : il permet la recherche d'une séquence de caractères dans des fichiers AppleWorks sur disque.

Vous avez bien lu : la recherche porte sur le contenu des fichiers. L'application est donc d'une grande puissance, à vous de savoir bien la manier. La prospection se fait d'ailleurs à une vitesse surprenante, si l'on tient compte du fait que le contenu entier du fichier est passé en revue.

La recherche en cours peut être nuancée : comme pour la fonction ⌘-L d'AppleWorks, vous pouvez choisir "Texte brut" ou "Image exacte"; de plus, la chaîne indiquée peut être limitée —ou non— aux seuls mots entiers. Par exemple, si vous choisissez la chaîne "PAS", l'application

retrouvera selon le cas uniquement le mot "pas", ou aussi "passablement", "dépassé", etc.

En cours de recherche, une fenêtre d'information s'affiche au milieu de l'écran. Elle indique le déroulement de la prospection : nom du fichier prospecté, type de ce fichier, chaîne cherchée, emplacement de chaque chaîne trouvée (le contexte est affiché). On peut à tout moment poursuivre, charger le fichier sur le Bureau, ou arrêter. En fin de recherche, un petit bilan s'affiche dans la fenêtre : nombre de fichiers prospectés, et nombre de correspondances trouvées.

La grande classe, quoi. Et le grand confort !

Vérifier disque

Cette application travaille en deux modes : Lecture simple, ou Lecture/Écriture. Elle lit un disque entier bloc par bloc (et le réécrit au fur et à mesure dans la deuxième option). Vous pouvez l'interrompre à tout moment par Escape.

Si elle rencontre un bloc défectueux, elle le signale et fait une pause avant de continuer.

À employer si vous n'arrivez pas à charger un fichier AppleWorks. Si aucune erreur n'est détectée, l'erreur est dans le libellé du fichier ou du catalogue. Si une erreur est détectée, le défaut est dans la disquette elle-même, ou dans les données du bloc. De toute façon, ce n'est pas drôle, mais un homme prévenu en vaut deux...

Voir Presse-papiers

Indique le type des données du presse-papiers, et montre son contenu à l'écran.

Particulièrement utile si vous employez l'application Triple Presse-papiers de Randy Brandt, et n'êtes plus tout à fait sûr du contenu de l'un des presse-papiers : maintenant vous pouvez vous en assurer sans avoir à essayer de le copier dans un fichier (avec toutes les manipulations et fausses manœuvres qui vous menacent si vous tentez de copiez un presse-papiers dans un fichier de type différent...).

Non seulement le contenu du presse-papiers peut être consulté, mais il peut être sélectivement effacé (ligne à ligne pour du TdT, fiche à fiche pour une BdD, ligne ou colonne pour un Tableur).

Des Ascii et des pommes

Dimitri Geystor

Je vous propose aujourd'hui d'explorer certains comportements d'AppleWorks, et d'UltraMacros.

Il sera question surtout de lecture et d'affichage de caractères à l'écran, dans le traitement de textes.

À titre d'illustration, je montrerai comment l'on peut, sans trop de peine, créer avec UltraMacros de jolies fenêtres de messages à l'écran, avec caractères normaux, MouseText et même en inverse vidéo. (La disquette d'accompagnement optionnelle Pom's 44 comporte le fichier `Macro.Fenêtre` prêt à l'emploi).

Pour travailler à l'aise je vous recommande d'avoir sous la main le fichier de macros `SuperPrint.um` de Daniel Lurot (Pom's, n° 42), la disquette PathFinder de Randy Brandt (et plus précisément l'application TimeOut "Valeurs ASCII" qu'elle contient). Ceci n'a rien d'obligatoire mais, à moins d'être masochiste, pourquoi travailler, comme on dit, avec "une main attachée derrière le dos" ?

En cours d'article, j'expliquerai aussi pas-à-pas comment utiliser l'utilitaire Block.Warden de ProSel – à l'intention de ceux d'entre vous qui possèdent ProSel, l'emploient pour lancer des programmes, mais n'ont jamais osé s'en servir pour "éditer" un fichier. Il n'y a rien de sorcier là-dedans.

Je vous recommande enfin de transformer vos fichiers SuperPrint.um en fichier Task, selon la méthode décrite dans mon précédent article, au moyen de l'application Macro Options. Ainsi, vous pourrez y accéder instantanément en cas de besoin. Quant aux petites macros qui émaillent l'article, avant le macro-programme proprement dit, vous pouvez soit les compiler toutes d'un coup (aucune ne porte le même nom), soit les compiler au fur et à mesure de votre lecture.

Bon, j'en viens à mon propos. Faisons un peu d'exploration, en commençant par le plus banal.

Le clavier

Quand vous appuyez sur une touche, ou une combinaison de touches, un signal est envoyé à votre ordinateur sous la forme d'un code, dit code ASCII. C'est un nombre, de 0 à 255 en notation décimale, ou de $00 à $FF en notation hexadécimale.

Le manuel d'UltraMacros contient, en dernière partie, une table des valeurs ASCII qu'il est commode de consulter d'un seul coup d'œil. Vous constaterez tout de suite que les caractères sont au nombre de 128 seulement, deux valeurs étant attribuées à chacun d'entre eux, une valeur "basse", et une valeur "haute". La différence entre l'une et l'autre est toujours de 128, ce qui correspond justement à la valeur du huitième bit d'un octet.

Si vous avez "Valeurs ASCII" sur votre menu TimeOut, appelez-le, et parcourez les divers codes en promenant la souris de gauche à droite. Puis faites "Return" deux fois, et tapez sur une touche ou une combinaison de touches : les codes correspondant s'afficheront aussitôt.

L'écran

Votre Apple II profite de cette double attribution de codes pour afficher les caractères de deux façons différentes : le mode "ordinaire" et le mode "inverse". Si, de plus, votre Apple dispose des ROMS qui accompagnent le processeur 65C02 (c'est le cas de toutes les machines récentes) il vous est possible d'afficher à l'écran un jeu de caractères supplémentaires "MouseText". Si vous programmez en Basic Applesoft vous savez qu'il faut, pour activer et désactiver ce jeu de caractères, envoyer des codes de contrôle spéciaux.

Les choses se passent autrement avec AppleWorks.

Comment AppleWorks interprète-t-il le code envoyé par le clavier ? Et qu'affiche-t-il à l'écran ? AppleWorks fait un double filtrage à sa manière, de telle sorte qu'il est impossible à l'utilisateur de taper directement des caractères en Inverse ou MouseText. Ces caractères ne sont accessibles qu'à travers des routines internes.

Pour toutes les valeurs ASCII basses comprises entre 32 (Espace) et 126 (Tréma), AppleWorks affichera à l'écran le caractère "normal" correspondant à (ASCII + 128). La valeur 127 correspond au "Delete".

Les valeurs "hautes" sont, pour AppleWorks, des commandes "Ó-". Il y réagira s'il les reconnaît, ou enverra un "BIP" si elles lui sont inconnues.

Les commandes UltraMacros liées aux codes ASCII

Key

UltraMacros possède une commande qui permet de lire la touche qui a été enfoncée. C'est la commande <key>. Exemple :

```
k:<awp
    msg " Appuyez sur une touche ou un
        e combinaison de touches ":
    k=key      (attend qu'on appuie sur
        une touche)
    msg k>!    (la valeur de la touche
        est dans la variable k)
```

Quand vous actionnez cette macro, et que vous appuyez ensuite sur une touche ou une combinaison de touches, elle vous renvoie en message de bas de ligne le code ASCII correspondant, tel qu'envoyé par le clavier et reçu par AppleWorks. Essayez par exemple de taper "M", et la valeur renvoyée sera 77 en décimal. C'est la valeur ASCII basse, ou valeur ASCII avec le huitième bit à zéro. Essayez maintenant la macro avec " ⌂-M". La valeur renvoyée sera 205 , soit 77 + 128. C'est la valeur ASCII haute, ou la valeur ASCII avec le huitième bit à 1 (le huitième bit vaut 128).
Compte-tenu de ce qui précède, vous voyez que <key> lit la touche telle qu'AppleWorks l'interprète.

Print

Au lieu de taper un caractère au clavier, on peut l'envoyer à AppleWorks par la commande <print>. Exemple :

```
p:<awp
    print "M">! (élémentaire, mon cher
        Watson)
```

On peut aussi d'abord définir une chaîne, puis imprimer la chaîne :

```
q:<awp
    $1="M"        (on définit la chaîne)
    print $1>!    (on l'imprime)
```

Chr$, Asc $, Str$

Dans la macro ci-dessus, on aurait pu définir la chaîne à partir d'une valeur ASCII. L'intérêt d'une telle définition est qu'on peut envoyer à AppleWorks des commandes "⌂-quelque-chose" :

```
r:<awp
    $1=chr$ 177   (177 = code ASCII pou
        r oa-1)
    print $1>!    (on envoie la command
        e à AppleWorks)
```

Elle permet aussi de transmettre des touches. En général, on ne change pas la valeur de k. Mais si l'on s'amuse, par exemple, à incrémenter k d'une unité, cela donne :

```
f:<awp (Clavier fou. Quitter avec Esc)
    begin
    k=key ifnot k=27
    k=k+1 print chr$ k : rpt
    else stop >!
```

Actionnez cette macro, et essayez de taper du texte...

Revenons à la macro sa-r, et compliquons-la un peu, en lui faisant donner le code ASCII du caractère imprimé (la commande <asc $> lit la valeur ASCII du premier caractère d'une chaîne) :

```
s:<awp
    $1="M"        (on définit la chaîne)
    print $1 :    (on l'imprime)
    k=asc $1      (la variable k contien
                  t le code ASCII de $1)
    msg k  >!     (k est imprimé en mess
                  age de bas de ligne)
```

La commande <str$>, quant à elle, transforme une variable numérique en chaîne de caractères. La macro ci-après imprime la caractère M et sa valeur ASCII :

```
t:<awp
    $1="M"        (on définit la chaîne)
    k=asc $1      (la variable k lit le
            code ASCII de $1)
    $2 = str$ k (on transforme la vari
            able numérique en chaîne)
    print $1 + " = " + $2>! (on imprim
            e 3 chaînes "concaténées")
```

Peek $10f5

Cette commande analyse le caractère sous le curseur (dont AppleWorks conserve le code à l'adresse hexadécimale $10f5) et renvoie la valeur de ce code ASCII dans la variable numérique désignée. Exemple :

```
c:<awp
    k=peek $10f5 (envoie le contenu de
            $10f5 dans la variable k)
    msg k>! (affich k en bas de ligne)
```

Si le curseur était sur un "M", surprise, ce n'est pas 77 mais 205 (77 + 128) qu'affiche le message. Eh oui, c'est ainsi, autant le savoir : la commande <peek $10f5> renvoie la valeur haute du code ASCII du caractère sous le curseur.

Poke $10f5,x

Là, nous nous aventurons un peu. L'hypothèse étant : si <peek $10f5> lit la valeur du caractère sous le curseur, pourquoi ne pas bluffer AppleWorks en mettant une quelconque valeur ASCII dans cette mémoire ? Essayons :

```
m:<awp
    k=193         (valeur haute de "A")
    poke $10f5,k>!
```

Ah, mais ça a l'air de marcher ! On actionne la macro, et un "A" apparaît sous le curseur. Hélas, c'eût été trop beau. On a bien mis un caractère à l'écran, mais en prenant un raccourci, et AppleWorks n'est malheureusement pas au courant. À la première occasion, il réécrira par-dessus ce que lui, AppleWorks, est censé mettre à cet endroit. Faites Escape deux fois de suite, et pfuiit ! votre caractère de contrebande est envolé.

Ne nous décourageons pas, et poursuivons notre exploration : du moment que `<peek $10f5>` envoie la valeur haute du code ASCII, que se passera-t-il si nous faisons un `<poke $10f5>` avec la valeur BASSE ? Essayons :

```
v:<awp
     k=65 (valeur basse de "A")
     poke $10f5,k>!
```

Et voilà une jolie pomme. On ne peut pas l'effacer par Delete, mais elle disparaît si l'on écrit par dessus (même en mode insertion) et, comme tout à l'heure, elle s'évanouit définitivement au moindre "rafraîchissement" d'écran (par Escape ou défilement).

Essayons de la capturer en la mettant dans la macro zéro ($0) avec la commande Ó-/ : aïe, c'est "A" qui nous revient, bien permanent cette fois, mais ce n'est pas ce que nous voulions.

Essayons de la copier avec Ó-C : elle s'envole aussitôt, avant même qu'on ait pu la saisir.

Et pourtant, n'y a-t-il pas quelque chose à tirer de cette nature fugace ? Si seulement, au lieu d'afficher un seul caractère, on pouvait en afficher plusieurs, on tiendrait là une façon de mettre à l'écran (à l'endroit du curseur) des messages "propres", qui disparaîtraient ensuite sans laisser de traces en restituant le texte qu'ils auraient éventuellement oblitéré.

Eh bien, ça marche sans problème. Par exemple :

```
u:<awp
     k=64
     begin
     poke $10f5,k right
     k=k+1 ifnot k=96 rpt else stop>!
```

Mettez le curseur en début de ligne, et tapez sa-u pour voir s'afficher les caractères MouseText.

De là, cependant, à pouvoir afficher commodément un message dans une fenêtre, avec la possibilité de modifier facilement et le message, et la position de la fenêtre, il y avait plus qu'un pas à franchir. C'est ce que fait le macro-programme que je vous présente en fin d'article.

Mais avant de l'aborder je ferai un détour par Block.Warden, l'éditeur d'octets de ProSel, et pour ceux qui ne l'auraient pas, par `SuperPrint.um` (les macros de Daniel Lurot du n° 42 de Pom's). Enfin, ceux qui n'ont même pas les macros SuperPrint.um trouveront quand-même en fin d'article la macro sa-$ de Daniel Lurot, qu'ils pourront recopier soigneusement à la main, compiler et utiliser pour introduire de façon permanente des caractères spéciaux (MouseText ou Inverse) dans un fichier de traitement de textes AppleWorks.

En effet, s'il fallait par la méthode des `<poke $10f5>` créer une fenêtre (cadre composé en partie de caractères MouseText, et en partie de caractères ordinaires), et y mettre un texte qui contienne au choix des caractères ordinaires, MouseText ou inverses, la tâche serait fort laborieuse. De plus, il serait très difficile de modifier le texte du message.

Or nous savons (depuis le n° 42 de Pom's), que des caractères spéciaux peuvent être introduits de façon permanente dans un fichier TdT AppleWorks, et qu'alors ils sont copiables à volonté par la commande Ó-C. J'ai constaté en outre qu'ils étaient lus correctement dans tous les cas par la commande `<asc $>`. Il est donc possible de créer d'avance des chaînes composées de caractères mixtes, qui serviraient de modèle à celles du message à afficher ; il suffirait ensuite d'écrire une routine qui les lirait, caractère par caractère, avec la commande `<asc $>`. Créer ces caractères avec SuperPrint.um est un jeu d'enfant, je n'y reviendrai pas. Je dirai quelques mots par contre de l'emploi de Block.Warden, de ProSel. Si vous vivez (heureux ?) sans ProSel, sautez la partie qui lui est consacrée.

Tant que vous êtes dans AppleWorks, créez un petit fichier de Tdt que vous nommerez Essai. Tapez-y trois fois de suite la ligne suivante, en changeant le numéro du début :

```
(1)  à A B C D E F G H I J K L M N O P Q R S T
U V W X Y Z ° ç ] ^  _
```

(avec un espace entre chaque caractère, et une ligne vide entre chaque ligne).

Sauvez Essai, quittez AppleWorks. Vous êtes maintenant dans l'écran sélecteur ProSel.

ProSel : À partir de l'écran de sélection principal, tapez les deux chiffres qui correspondent au Port et au Lecteur où se trouve la disquette qui contient le fichier Block.Warden. Par exemple 51 pour le Port 5, Lecteur 1.

Vous verrez s'afficher une liste de fichiers-programmes (en majuscules) et de sous-catalogues (en minuscules). Si Block.Warden n'est pas visible, sélectionnez le sous-catalogue qui le contient (probablement "util").

Sélectionnez et validez Block.Warden. Il faut maintenant lui indiquer où est votre fichier Essai. Changez au besoin de disquette (Block.Warden reste en mémoire), et indiquez le préfixe ProDOS. Pour cela tapez sur la touche "P" : une fenêtre apparaît, avec un préfixe par défaut qui n'est sans

doute pas le bon. Tapez le préfixe voulu, en commençant bien par une barre oblique (/).

Par exemple `/travail/tdt/`. Faites Return, et le préfixe actuel s'inscrira à gauche en haut de l'écran. Maintenant, tapez "F" pour appeler le fichier. Une fenêtre s'ouvre, avec "Follow/Info pathname". Tapez le nom du fichier, sans barre oblique. Par exemple "Essai". Faites Return.

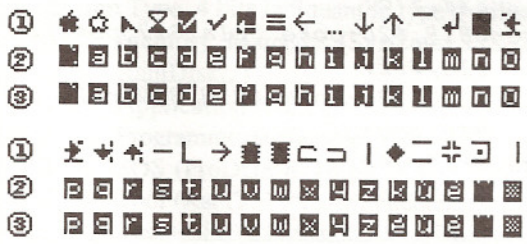
L'écran change aussitôt, et vous devriez y voir déjà des choses familières. Cet écran est divisé en deux parties : à gauche, les valeurs hexadécimales de tous les octets du fichier, à droite leurs équivalents en caractères ASCII. En haut à droite de l'écran, la mention "R/W mode", c'est à dire mode Lecture/Écriture. On circule écran par écran, avec les flèches haut/bas. Un bloc occupe deux écrans.

Tapez sur la touche "E". En haut à droite, la mention est maintenant "Edit mode". On circule caractère par caractère avec les flèches haut/bas et droite/gauche, et l'on fait passer le curseur au choix dans la partie droite ou gauche de l'écran avec la touche TAB. On ne peut éditer qu'un bloc à la fois (deux écrans). Pour revenir au mode R/W, taper Escape. Pour sauver les changements, taper W pour "write" suivi de Return, et valider avec la touche "Y".

Repérez, dans la partie droite de l'écran, les trois lignes de caractères que vous aviez tapées dans AppleWorks. Mettez le curseur sur le "à" de la première ligne, et faites TAB : le curseur saute à gauche sur le code hexadécimal correspondant, qui doit être 40. Tapez c0 (c'est à dire un c à la place du 4 et aussitôt après un zéro). Le curseur est maintenant sur la valeur 20, qui est le code hexadécimal de l'espace. Avec la flèche à droite, allez sur la valeur suivante qui est 41, et tapez c1. Continuez ainsi, en remplaçant les 4 par des c, et les 5 par des d, jusqu'au dernier caractère de cette ligne du fichier AppleWorks, qui est 5f, et qui doit devenir df. Faites Esc pour revenir au mode R/W, tapez W et Return, suivi de Y. Le disque tourne, les modifications que vous avez apportées ont été enregistrées.

Revenez au mode "Edit" avec la touche "E", et modifiez les deuxième et troisième lignes de la même manière, mais en remplaçant les 4 et 5 par des e et f dans la 2ème ligne, et par des a et b dans la 3ème (donc, de e0 à ff, et de a0 à bf). Sauvez, et quittez Block.Warden en tapant "Q" suivi de "Y" pour confirmer.

Lancez AppleWorks, chargez le fichier Essai, et admirez votre œuvre. Si vous n'avez pas fait de faute, les trois lignes sont devenues :



En dépit des apparences, les lignes (2) et (3) ne sont pas identiques. Mais il est impossible de distinguer facilement, à partir d'AppleWorks, les caractères des séries (2) et (3). Une solution est de recopier le caractère à tester dans la macro suivante, de la compiler et de l'exécuter ; elle affichera le caractère spécial en inverse vidéo, et indiquera sa valeur ASCII en bas d'écran.

```
w:<awp
      $1="■" (définition de la chaîne)
      v=asc $1 (lecture de la valeur ASC
            II du 1er caractère)
      n=v+128 (on inverse le 8ème bit)
      if n>255 n=n-256 elseoff
      poke $10f5,n (on poke la nouvelle
            valeur)
      msg n>! (et on l'affiche aussi en
            bas d'écran)
```

Voici les caractères qui correspondent à la série (3) :

(3) ! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?

Devant le point d'exclamation, il y a un espace (non visible).

Nous voici maintenant prêts à construire notre macro-programme.

La première tâche est d'écrire les chaînes qui formeront la fenêtre. Elles logeront dans la sous-routine ba-2. Tapez ces chaînes au clavier, directement pour les caractères "normaux", et avec la macro sa-$ de Daniel Lurot (donnée en fin d'article). Vous pouvez aussi copier caractère par caractère à partir des trois lignes obtenues avec Block.Warden. Si vous avez essayé l'un et l'autre, vous ne douterez pas de la supériorité écrasante de la macro.

Pour changer le texte du message, il suffit de modifier les chaînes. Pour déplacer la fenêtre, il suffit de changer les coordonnées dans la macro ba-1.

La macro de démarrage ba-1 positionne le curseur au coin supérieur gauche de la future fenêtre, puis passe aux sous-routines.

Note 1 : Les routines ba-3 et ba-4 utilisent la notation $(k) pour désigner les chaînes, où k est une variable que l'on incrémente : notation très pratique quand une série de chaînes doivent subir le même traitement.

Note 2 : La partie `if n > 255 n = n - 256 elseoff` de la macro sa-v ci-dessus a été omise : on peut s'en passer, AppleWorks rétablit la valeur ASCII dont il a besoin. Dans la routine ba-4 on peut aussi mettre v=v-128, le résultat est le même.

Macro.Fenêtre

```
START

Macro principale :
------------------

<ba-1>:<awp (repérages et mise en place
          du curseur)
     insert (mode insertion pour un cur
          seur moins voyant)
     Ó-up ba-left (curseur en haut à ga
          uche de l'écran)
     posn x,y (on note sa position)
     x=x+25 (coordonnées du coin supéri
          eur gauche...)
     y=y+6 (...de la future fenêtre)
     begin posn a,b down b=b+1 ifnot b=
          y rpt else (on descend de 6)
     begin posn a,b right a=a+1 ifnot a
          =x rpt else (on avance de 25)
     ba-2 (lecture des chaînes qui cons
          tituent la fenêtre)
     ba-3>! (exécution de la fenêtre)

Routines :
----------

<ba-left>:<awp (renvoi du curseur en dé
          but de ligne)
     c=0
     begin Ó-tab c=c+1 ifnot c=18 rpt>!

<ba-2>:<awp (Définition des chaînes qui
          feront la fenêtre)

    $1="________________________________"
    $2="|      Exemple : Message       |"
    $3="|       en caractères          |"
    $4="|     normaux, MouseText       |"
    $5="|         et INVERSE           |"
    $6="|______________________________|"
     >!  (Fermons la macro)

<ba-3>:<awp (boucle d'appel des chaînes
          successives)
     k=1 (k=1 pour la première chaîne)
     begin
     ba-4 down ba-5 k=k+1 ifnot k=7 rpt
          else down stop>!

<ba-4>:<awp (boucle d'affichage tempora
ire d'une chaîne)
     L=len $(k) (longueur de la chaîne)
     v=asc $(k) (lit la valeur ASCII du
          premier caractère)
     v=v+128(on rétablit le 8ème bit)
     poke $10f5,v right  (on poke et on
          avance)
     L=L-1 ifnot L=0 $(k)=right $(k),L
          (on fait sauter le premier ca
          ractère de la chaîne)
     rpt>! (et on recommence la boucle)

<ba-5>:<awp (curseur à la ligne, et au
          bord gauche)
     left posn x,y if x>a rpt>!
```

Macros de saisie de caractères spéciaux

(Daniel Lurot, Pom's n°42)

```
$:<awp msg 'Tapez le caractère désiré':
      A = key msg '':
      if A<32 then A=32+A elseoff
      print chr$ A><
      left:goto ba-M>!

<esc>:<awp>;<left:goto ba-M>!
<ctrl-v>:<awp>6<left:goto ba-M>!
<ba-M>:<awp><
     poke $800,$8d:poke $801,$08:poke $
          802,$c0:
     poke $803,$20:poke $804,$1d:poke $
          805,$6f:
     poke $806,$c9:poke $807,$81:
     poke $808,$d0:poke $809,$12:
     poke $80a,$a5:poke $80b,$82:
     poke $80c,$38:
     poke $80d,$ed:poke $80e,$00:poke $
          80f,$7c:
     poke $810,$a8:
     poke $811,$b9:poke $812,$02:poke $
          813,$7c:
     poke $814,$49:poke $815,$80:
     poke $816,$99:poke $817,$02:poke $
          818,$7c:
     poke $819,$20:poke $81a,$17:poke $
          81b,$6f:
     poke $81c,$8d:poke $81d,$09:poke $
          81e,$c0:
     poke $81f,$60:
     call $800><oa-Z oa-Z>!
```

Un nouveau CDA GS : Ram.Use

Patrice Neveu

Seize millions sept cent soixante-dix sept mille deux cent seize octets, c'est ce que le 65SC816 de votre Apple IIGS peut adresser, soit 256 fois plus que ce dont était capable l'ancestral 6502. Pour gérer ces 16 Mo, le GS est doté d'une gestion de mémoire performante appelée Rom Tool 02, ou Memory Manager.

Le Memory Manager

Dans le principe, le rôle principal du Memory Manager est de comptabiliser la mémoire de telle façon qu'en l'utilisant, un programme peut allouer ou désallouer des blocs de mémoire ainsi que changer leur taille pour satisfaire ses besoins. Le Memory Manager prend note de chaque utilisateur de chaque bloc de mémoire, afin de ne jamais donner plus d'une fois une zone de mémoire.

Pratiquement, pour connaître à chaque instant l'état de la mémoire, le Memory Manager entretient une liste lui indiquant pour chaque bloc :

- l'adresse de début du bloc ;
- le statut du bloc, (s'il est dynamique, effacable...) ;
- l'utilisateur du bloc ;
- la taille du bloc.

C'est cette liste que peuvent vous donner le Memory Mangler, CDA fourni avec la Documentation Développeur, et le Memory Peeker, CDA installé dans les ROMs version 01 (voir Pom's 35 page 63).

L'ID Tag Manager

Pour être capable de reconnaître chaque utilisateur, le Memory Manager travaille avec l'ID Tag Manager, qui fait partie des Miscellaneous Tools (Tool 03). L'ID Tag Manager est utilisé pour assigner un identificateur (ID) à chaque programme en fonction de son type. Cet ID est composé de trois champs :

① Champ Type, 4 bits indiquant le type d'utilisateur :
- 0 = Memory Manager (Système, E/S, Affichage, RamDisk...)
- 1 = Application (GS.Paint, APW...)
- 2 = Programme de contrôle
- 3 = SOS (ProDOS 8, 16 ou GS.OS...)
- 4 = Tool (RamTools essentiellement)
- 5 = Accessoire de bureau (CDA, NDA)
- 6 = Bibliothèque (fichier RTL)
- 7 = System Loader
- 8 = Fonction du Firmware ou du Système
- 9 = Tool Locator (RomTool 01)
- A = Mise à jour (fichier STR)
- B à F = Non-défini

② Champ Type Auxiliaire, 4 bits définis par l'utilisateur.

③ Champ d'identification, 8 bits assignés par l'ID Tag Manager.

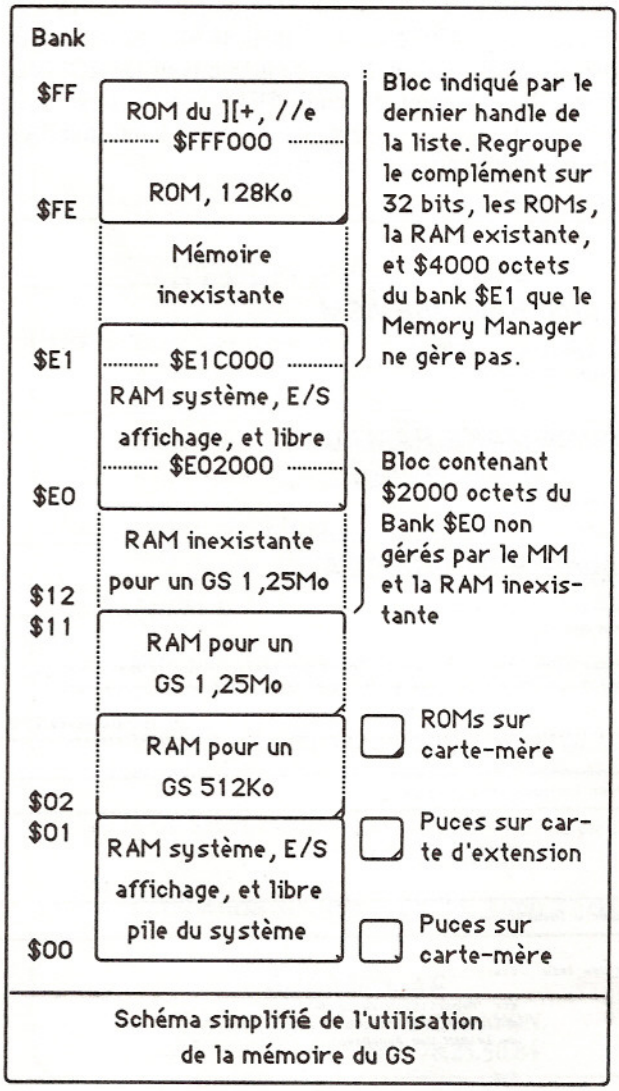


Schéma simplifié de l'utilisation de la mémoire du GS

En pratique

Question : Comment se répartissent tous ces éléments dans ma mémoire ?

Réponse : Faites ♂-Ctrl-Escape, selectionnez le CDA "Répartition de la Mev", et conformez-vous aux indications inscrites en bas de l'écran du CDA.

Question : J'en suis à l'affichage en pourcentage. Qu'en est-il ?

Réponse : Les lignes 'MEV utilisée' et 'MEV libre' sont en poucentage de la Mev Totale. Les lignes Memory Manager, Applications jusqu'à Non-Défini sont en pourcentage de la MEV utilisée. La ligne RAM Disk est en pourcentage du Memory Manager.

Au premier coup d'œil sur le schéma simplifié de la mémoire du GS, on a l'impression que tout est bien organisé. Certes, mais en fait, il ne faut pas oublier que ce schéma est simplifié à l'extrême et qu'en outre, deux particulatités, signalées par les accolades y sont remarquables.

Pour expliquer le schéma, il faut dire qu'il y a une importante zone vide en plein dans la MEV, ainsi qu'entre les banks $E0- E1 et la ROM. Pour ne pas allouer le trou au milieu de la MEV à un utilisateur, le Memory Manager considère qu'il l'utilise. Il fait de même pour les ROMs et l'espace les séparant du bank $E1. Allez-voir avec le Memory Peeker : vous trouverez deux handles correspondants à des blocs énormes.

Malheureusement, pour ce faire le Memory Manager n'utilise pas un bloc fictif allant de la fin de la MEV étendue jusqu'au début du Bank $E0, mais un bloc allant de la fin de la MEV étendue jusqu'à la page $20 du Bank $E0 ($E02000), car les $20 premières pages du bank $E0 sont réservées au système, et ne sont ainsi pas gérées par le Memory Manager. En conséquence, ce dernier mélange allègrement de la MEV réelle et de la MEV inexistante.

De même, parce qu'il est impossible d'allouer une zone en ROM ou dans l'espace actuellement vide situé entre le bank $E1 et la ROM, le Memory Manager crée non pas un bloc allant de la fin du bank $E1 jusqu'en $FFFFFF (24 bits), mais un bloc partant de $00E1C000 jusqu'en $FFFFFFFF (32 bits), car la zone $E1C000 à $E1FFFF est elle aussi réservée au système (E/S et carte langage).

Remarquez qu'en considérant que la mémoire se termine en $FFFFFFFF, Apple se réserve ainsi une porte facile vers la compatibilité avec un processeur adressant 32 bits.

Ces deux trous dans la mémoire sont bien évidemment pris en compte par le présent CDA, ainsi que les $6000 ($2000+$4000) octets réels 'mélangés' à la mémoire absente.

Amusez-vous

Il est très intéressant d'utiliser ce CDA sous différents environnements tels que APW, AppleWorks, GSPaint, un Quit ProDOS (8 et 16), l'AppleSoft, ProDOS 8, ProDOS 16, GS.OS...

Source RAM.USE.ASM

```
Asml RAM.Use.CDA keep= RAM.Use
Filetype RAM.Use CDA
Exit
```

Source MAKE.RAM.MAC

```
macgen RAM.Use.CDA RAM.Macros /RAM5/Macros/=
RAM.Use.ASM
Exit
```

Source RAM.USE.CDA

```
*****************************************************************************
* RAM.Use v1.6                                                              *
*                                                                           *
* Permet d'afficher l'utilisation de la memoire geree par le Memory Manager,*
* selon les IDs definis dans le ID Tag Manager des Miscellaneous Tools.     *
*                                                                           *
*                                                  (c) Patrice Neveu 1988   *
*****************************************************************************

*---------------------------------------------------------------------------*
* Pour l'assemblage / linkage                                               *
*---------------------------------------------------------------------------*

            mcopy RAM.Macros
            keep RAM.Use

*---------------------------------------------------------------------------*
* Bloc d'identification de CDA                                              *
*---------------------------------------------------------------------------*

RAM_Use_Info  Start

            str 'Repartition de la MeV'
            dc i4'RAM_Use_StartUp'
            dc i4'RAM_Use_ShutDown'

            End

*---------------------------------------------------------------------------*
* Segment principal du CDA                                                  *
*---------------------------------------------------------------------------*

RAM_Use_StartUp Start
            using RAM_Use_Data

WndLft      equ $20
WndWdth     equ $21
WndTop      equ $22
WndBtm      equ $23

First_Handle  gequ $00E11600

*---------------------------------------------------------------------------*
* Prepare le DataBank                                                       *
*---------------------------------------------------------------------------*

            phb                 Sauve le banc de donnees de l'appeleur
            phk                 Car a sa place on met le notre, celui
            plb                 De notre banc de programme

*---------------------------------------------------------------------------*
* Demande de combien on dispose de RAM, calcule la fin de la RAM et la RAM  *
* utilisee                                                                  *
*---------------------------------------------------------------------------*

            PushLong £0
            _TotalMem
            PullLong RAM_Totale

            ldx RAM_Totale
            ldy RAM_Totale+2
            stx EORam
            dey                 Retire les blocks E0 & E1, pour savoir
            dey                 Se trouve le trou en RAM
            sty EORam+2

            PushLong £0         Demande combien de RAM est libre
            _FreeMem
            PullLong RAM_Libre

            lda RAM_Totale      RAM totale - RAM libre = RAM utilisee
            sec
            sbc RAM_Libre
```

```
            sta RAM_Utilisee
            lda RAM_Totale+2
            sbc RAM_Libre+2
            sta RAM_Utilisee+2

*-----------------------------------------------------------------------*
* Prepare le mode video et les marges qui lui conviennent au cadre     *
*-----------------------------------------------------------------------*

            PushWord £0             Place pour le resultat
            PushWord £$19           Mode video de la Battery RAM
            _ReadBParam
            lda £$2800
            plx
            beq Video40
            ora £$0014
Video40     sta >$0020

*-----------------------------------------------------------------------*
* Affichage du cadre CDA, et mise a jour des marges a l'interieur du cadre *
*-----------------------------------------------------------------------*

            PushLong £Cadre
            _WriteCString
            shortm
            lda £38
            sta >WndWdth
            lda >WndLft
            ina
            sta >WndLft
            lda £5                  WndTop= 6 & WndBtm= 21
            sta >WndTop
            lda £21
            sta >WndBtm
            longm

*-----------------------------------------------------------------------*
* Calcule les totaux par categories d'ID                                *
*-----------------------------------------------------------------------*

***     Initialise les variables.........................................

            ldy £4*12-4             Met toutes les Sizes a 0
            lda £0
Clear_Sizes sta Sizes,y
            dey
            dey
            bpl Clear_Sizes

            stz RAMDisk_Size        RAMDisk a 0
            stz RAMDisk_Size+2

            stz Mode                Mode 0= affichage des tailles en octets

            ldx £<First_Handle      Passe le 1er handle
            ldy £^First_Handle
            stx Handle
            sty Handle+2

            PushLong Handle         Transforme le handle en adresse
            jsr HndlToAdr
            Pull3 Adresse+1         Depile l'adresse
            shortm
            pla                     Depile le 4eme octet (nul)
            longm

Next_Maillon ldy £6                 Index pour lire l'ID
            jsr Read_Maillon
            sta ID

***     Indexation pour la table des tailles.............................

            and £$F000              Garde le Type Field (nibble 4)
            lsr a                   Passe-le dans le nibble 3 (4321)
            lsr a
            lsr a
            lsr a
            xba                     Passe en bas (nibble 1)
            cmp £$B                 Au dessus du type $A (SetUp File), tout
            bcc Indexe              Est considere comme $B (Non-Defini), et
            lda £$B                 Ce, afin que l'affichage se fasse bien.
Indexe      asl a                   x2
            asl a                   x4
            tax                     Index pour mettre dans la bonne somme

;       *-------------------------------------------------------*
;       * Cas special du Type 0 (Memory Manager) qui peut etre: *
;       * ------------------------------------------------------ *
;       *       Memory Manager: type $0xxx / Statut $xxxx       *
;       *       RAMDisk       :      $0001          $0008       *
;       *       False RAM            $0000          $C000       *
;       *-------------------------------------------------------*

            bne Read_Length

Type_Nul    lda ID
            beq Full_MM
            cmp £$0001              RAMDisk
            bne Read_Length

* Cas du RAMDisk

            ldy £$4
            jsr Read_Maillon
            cmp £$0008              Statut des blocs du RamDisk
            bne Full_MM
            ldy £$8
            jsr Read_Maillon
            clc
            adc RAMDisk_Size
            sta RAMDisk_Size
            php
            ldy £$A
            jsr Read_Maillon
            plp
            adc RAMDisk_Size+2
            sta RAMDisk_Size+2

            bra Read_Length

***     Tester si l'on a atteind la limite de la Ram situee avant le bank $E0

Full_MM     ldy £0                  Lit l'adresse du bloc
            jsr Read_Maillon
            cmp EORam               Ca peut etre la fin de RAM ?
            bne Read_Length
            ldy £2
            jsr Read_Maillon
            cmp EORam+2
            bne Read_Length
            beq Try_Next            Fin de RAM, alors passe

Read_Length ldy £8                  Index pour lire la longueur
            jsr Read_Maillon
            clc
            adc Sizes,x
            sta Sizes,x
            php                     Conserve la retenue
            ldy £$A
            jsr Read_Maillon
            plp                     Recupere la retenue
            adc Sizes+2,x
            sta Sizes+2,x

***     Passe au prochain maillon.......................................

Try_Next    ldy £$10
            jsr Read_Maillon
            tax
```


APPLE EXPO STAND 1 I 8

VOS APPLE IIC
VOS APPLE IIE
VOS APPLE II GS.

N'EN REVIENDRONT PAS !

IIGS: Extensions mémoire (GS Ram et GS Ram Plus), Alimentation permanente (RamKeeper) Accélérateur 6.25 Mhz (TransWarp GS) Compatibilité MS-DOS avec PC-Transporter, Disque Dur interne 20/40/100 Mo Vulcan Co-processeur mathématique (FastMath) ...

IIE: Extension mémoire, Alimentation permanente Disque Dur interne 20/40/100 Mo Vulcan Disque dur winchester 3.5" externe 20 Mo Accélérateur 3,6 Mhz (TransWarp) etc ...

IIC: Extension mémoire avec ou sans horloge et CP/AM, UniDisk 800K et kit de mise à niveau.

TOUT LE MATERIEL APPLE II sur mesure: Apple II GS, LaserWriter, ImageWriter II,...

LES MEILLEURS LOGICIELS pour Apple II à des prix U.S.A dont en exclusivité:

AppleWorks GS, HyperStudio, AppleWorks 3.0

BREJOUX.AE
29 A rue Montribloud 69009 LYON
Tel: 78.36.52.69 Fax: 78.25.50.84

```
                ldy £$12
                jsr Read_Maillon
                shortm
                sta Adresse+3
                longm
                stx Adresse+1

* Verifie que ce maillon n'est pas le dernier

                ldy £$10
                jsr Read_Maillon
                cmp £0                  Necessaire car Read_Maillon se termine
                bne Go_Nxt_Maill        Par un plx qui change z
                ldy £$12
                jsr Read_Maillon
                cmp £0                  Voir les remarques precedentes
                beq Aff_Used_RAM

Go_Nxt_Maill    brl Next_Maillon

***     Affiche les tailles ainsi que la RAM utilisee et la RAM libre.........

Aff_Used_RAM    lda MM_Size             Corrige tout d'abord la taille
                clc                     Utilisee par le Memory Manager en
                adc £$6000              Ajoutant $6000 au total actuel
                sta MM_Size
                bcc No_Carry
                inc MM_Size+2

No_Carry        jsr Print_Liste         Affiche la liste de types

*-----------------------------------------------------------------------*
* Selection de l'utilisateur                                            *
*-----------------------------------------------------------------------*

***     Attend la frappe de touche, et aiguille ou il faut....................

Selectionne     jsr Read_KBd

                cmp £$1B                Escape= Quitter
                beq Sortie
                cmp £$20                Espace= Changer d'affichage
                bne Selectionne

***     Changement d'affichage...............................................

                inc Mode                Passe au mode prochain
                lda Mode
                cmp £3
                bcc Change_Aff
                stz Mode                Repasse a zero

Change_Aff      jsr Print_Liste         Affiche selon le nouveau mode
                bra Selectionne         Et retourne attendre l'utilisateur

***     Sortie...............................................................

Sortie          plb                     Recupere le banc de donnees
                rtl

                End

*-----------------------------------------------------------------------*
* Sous-Programmes:                  - Print_Liste                       *
*                                   - Read_KBd                          *
*                                   - Read_Maillon                      *
*                                   - HTab                              *
*                                   - Tab                               *
*                                   - HndlToAdr                         *
*-----------------------------------------------------------------------*

RAM_Use_Routines Start
                using RAM_Use_Data

*-----------------------------------------------------------------------*
* Affichage dans le mode demande: Mode 0= en octets                     *
*                                 Mode 1= en Ko                         *
*                                 Mode 2= en %                          *
*-----------------------------------------------------------------------*

Print_Liste     Entry

***     Affiche la RAM totale en mode inverse dans la partie noire du cadre...

                lda Mode                On n'affiche pas la RAM_Totale en %
                cmp £2
                beq Aff_Tailles

                lda £$011D
                jsr Tab
                PushWord £INV           Affiche en inverse la RAM totale
                _WriteChar

                MoveLong RAM_Totale,Bytes
                jsr Print_Size

                PushWord £NORM
                _WriteChar              Repasse en normal

***     Affiche les tailles..................................................

Aff_Tailles     MoveLong RAM_Utilisee,Quotient

                lda £$B
                sta Byte
Next_Size       ina                     Ajoute une ligne car on insere RamDisk

Bonne_Ligne     clc
                adc £3
                xba
                ora £$001D
                jsr Tab

                lda Byte
                asl a                   x2
                asl a                   x4
                tax
                MoveLong Sizes,x,Bytes
                jsr Print_Size

                dec Byte
                lda Byte
                bmi EOTypes             Si on depasse 0, alors termine
                bne Next_Size           Si on n'est pas encore a zero
                beq Bonne_Ligne         Si on est a zero

***     Affiche les totaux hors-liste........................................

EOTypes         MoveLong RAM_Totale,Quotient

                lda £$111D              Puis la RAM utilisee
                jsr Tab
                MoveLong RAM_Utilisee,Bytes
                jsr Print_Size

                lda £$121D              Affiche la RAM libre
                jsr Tab
                MoveLong RAM_Libre,Bytes
                jsr Print_Size

                lda £$0413              Affiche enfin le RamDisk
                jsr Tab
                MoveLong MM_Size,Quotient
                MoveLong RAMDisk_Size,Bytes
                jsr Print_Size

                rts

* Aiguillage vers les differentes sous-routines d'affichage

Print_Size      lda Mode
                beq Print_In_Bytes
                cmp £1
                beq Print_In_Ko
                bne Print_In_P

* Sous-routine d'affichage en octets

Print_In_Bytes  PushLong Bytes          Numerateur
                PushLong £Bytes_Str     Buffer
                PushWord £8             8 caracteres
                PushWord £0             Pas de signe
                _Long2Dec

                PushLong £Bytes_Str
                _WriteCString

                rts

* Sous-routine d'affichage en Ko

Print_In_Ko     PushLong £0
                PushLong £0
                PushLong Bytes
                PushLong £1024          Obtenir des Ko.
                _LongDivide

                PushLong £Quotient_Str_K
                PushWord £5             5 caracteres
                PushWord £0             Pas de signe
                _Long2Dec

                PushLong £Quotient_Str_K
                _WriteCString

                pla                     Retire le reste de la division
                pla

                rts

* Sous-routine d'affichage en %

Print_In_P      PushLong £0
                PushLong £0
                PushLong Bytes          Octets x100
                PushLong £100           Au maximum 16 Mo x100 tient dans 4 oct.
                _LongMul                Donc un Long suffit
                PullLong Bytes
                pla                     Depile la partie haute (nulle)
                pla

                PushLong £0
                PushLong £0
                PushLong Bytes
                PushLong Quotient
                _LongDivide

                PushLong £Quotient_Str_P
                PushWord £7
                PushWord £0
                _Long2Dec

                PushLong £Quotient_Str_P
```

```
            _WriteCString

            pla                     Retire le reste de la division
            pla

            rts

*-------------------------------------------------------------------------*
* Lecture du clavier                                                      *
*-------------------------------------------------------------------------*

KBd         equ $C000
KBdStb      equ $C010

Read_KBd    Entry
            lda >KBd                Attente d'une frappe au clavier
            bpl Read_KBd            On fait comme ca car malheureusement
            lda >KBdStb             _ReadChar derange les valeurs de Wnd...
            and £$007F              Masque le caractere
            rts

*-------------------------------------------------------------------------*
* Lecture d'un maillon                                                    *
*-------------------------------------------------------------------------*

Read_Maillon Entry

            phx                     Sauve X qu'il ne faut pas perdre
            tyx                     On doit faire cette gymnastique car
Adresse     Entry
            lda $112233,x           L'indexage en y sur 24 bits n'existe pas
            plx                     Recupere X
            rts

*-------------------------------------------------------------------------*
* HTab                                                                    *
*-------------------------------------------------------------------------*

HTab        Entry
            sta Byte_2
            lda >$0024
            and £$FF00              Efface la colonne actuelle,
            ora Byte_2

*-------------------------------------------------------------------------*
* Tabulation: VTAB, HTAB                                                  *
*-------------------------------------------------------------------------*

Tab         Entry
            sta >$0024

            lda £$FC22
            pha
            pha
            pha
            pha

            pha
            pha
            pha
            pha
            _FWEntry
            pla
            pla
            pla
            pla
            rts

*-------------------------------------------------------------------------*
* A partir du Handle qui est dans la pile, renvoie l'Adresse qui y correspond *
* dans la pile, en Long                                                   *
*                                                                         *
*-------------------------------------------------------------------------*

HndlToAdr   Entry
            php
            phx
            phy
            phb                     ; Sauve DB qui va etre modifie
            lda $B,s                ; Haut du handle
            shortm
            pha                     ; Passe le haut du handle ds le registre B
            plb                     ; Pour aller lire directement
            longm
            ldy £0
            lda (9,s),y             ; Lit le bas de l'adresse
            tax
            ldy £2
            lda (9,s),y             ; Lit le haut de l'adresse
            sta $B,s                ; A mettre dans la pile
            txa                     ; Passe le bas de l'adresse dans la pile.
            sta 9,s
            plb                     ; Recupere DB a l'entree
            ply
            plx
            plp                     ; Recupere P
            rts

            End

*-------------------------------------------------------------------------*
* Segment ShutDown du CDA                                                 *
*-------------------------------------------------------------------------*

RAM_Use_ShutDown Start
            using RAM_Use_Data
            rtl
            End

*-------------------------------------------------------------------------*
* Donnees                                                                 *
*-------------------------------------------------------------------------*

RAM_Use_Data  Data

*-------------------------------------------------------------------------*
* Codes de controle de l'affichage                                        *
*-------------------------------------------------------------------------*

CNORM       equ 24                  ; caracteres normaux
CSOUR       equ 27                  ;             souris
;
NORM        equ 14                  ; normal
INV         equ 15                  ; inverse
LF          equ 10                  ; descende d'une ligne
FF          equ 12                  ; equivalent de HOME
CR          equ 13                  ; revient en debut de ligne
EM          equ 25                  ; ramene le curseur en haut a gauche
FS          equ 28                  ; deplace d'un caractere a droite

BARH        equ 76                  ; L ($4C= 76) barre en haut
BARD        equ 90                  ; Z ($5A= 90) barre a droite
BARG        equ 95                  ; _ ($5F= 95) barre a gauche

*-------------------------------------------------------------------------*
* Definition des lignes pour le trace du cadre et de son contenu          *
*-------------------------------------------------------------------------*

Cadre       dc i1'FF,CNORM,NORM',c' ',38c'_',c' '
            dc i1'INV,CSOUR,BARD,CNORM,NORM',c' Utilisation de la MeV '
            dc i1'INV',c' Tot:          ',i1'CSOUR,BARG'

            dc i1'BARD',38i1'BARH',i1'BARG'

            dc i1'BARD,CNORM,NORM'
            dc c' Memory Manager_____________'
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'
            dc c'     dont RamDisk (        )'
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'
            dc c' Applications_______________'
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'
            dc c' Programmes de Controle_____'
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'
            dc c" Systeme d'exploitation_____"
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'
            dc c' Outils (RAM/ROM Tools)_____'
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'
            dc c' Accessoires de bureau______'
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'
            dc c' Bibliotheques (RTL)________'
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'
            dc c' System Loader______________'
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'
            dc c' Fonctions FirmWare/Systeme_'
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'
            dc c' Tool Locator_______________'
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'
            dc c' Mises a Jour (STR)_________'
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'
            dc c' Non-Defini_________________'
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'

            dc 38i1'FS',i1'INV,CSOUR,BARG,BARD,CNORM,NORM'

            dc c' MeV utilisee_______________'
            dc i1'CSOUR,INV,BARG,BARD,CNORM,NORM'

            dc c' MeV libre__________________'
            dc i1'CSOUR,INV,BARG'

            dc i1'BARD,CNORM,NORM',38i1'FS',i1'INV,CSOUR,BARG'

            dc i1'BARD,CNORM,NORM'
            dc c' Quitter: escape    Affichage: espace '
            dc i1'INV,CSOUR,BARG,CNORM,NORM'

            dc c' ',i1'INV,CSOUR',38i1'BARH',i1'CNORM,NORM',h'00'

*-------------------------------------------------------------------------*
* Variables                                                               *
*-------------------------------------------------------------------------*

ID          ds 2
Mode        ds 2

Byte        ds 2
Byte_2      ds 2
Handle      ds 4
Quotient    ds 4
Bytes       ds 4

Bytes_Str   ds 8
            dc h'00'
Quotient_Str_K ds 5
            dc c' Ko',h'00'
Quotient_Str_P ds 7
            dc c'%',h'00'

Sizes       anop
MM_Size     ds 4
            ds 4                    Applications
            ds 4                    Control Programs
```

```
                ds 4                    ProDOS
                ds 4                    Tools
                ds 4                    Desk Accessories
                ds 4                    Run Time Libraries
                ds 4                    System Loader
                ds 4                    FirmWare / System Functions
                ds 4                    Tool Locator
                ds 4                    SetUp Files
                ds 4                    Undefined

RAMDisk_Size    ds 4                    Taille du RAMDisk

RAM_Totale      ds 4                    Taille de la RAM installee dans le GS
EORam           ds 4                    Fin de la RAM avant le Bank $E0
RAM_Libre       ds 4                    Taille de la RAM installee restee libre
RAM_Utilisee    ds 4                    Taille de la RAM installee occupee

                End

*xxxx
```

```
2780:1C 1C 1C 1C 1C 1C 1C 1C 1C 1C 1C 1C 1C 1C 1C 1C
2790:1C 1C 1C 1C 1C 1C 1C 1C 0F 1B 5F 5A 18 0E 20 51
27A0:75 69 74 74 65 72 3A 20 65 73 63 61 70 65 20 20
27B0:20 20 41 66 66 69 63 68 61 67 65 3A 20 65 73 70
27C0:61 63 65 20 0F 1B 5F 18 0E 20 0F 1B 4C 4C 4C 4C
27D0:4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C
27E0:4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C
27F0:4C 4C 18 0E 00 00 00 00 00 00 00 00 00 00 00 00
2800:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2810:00 00 00 00 00 00 00 20 4B 6F 00 00 00 00 00 00
2820:00 00 25 F1 45 00 00 00 F5 04 00 16 00 1E 00 F5
2830:04 00 1A 00 E8 03 F5 02 00 30 00 13 08 F5 02 00
2840:34 00 15 08 F5 02 00 37 00 13 08 F5 02 00 3A 00
2850:15 08 F5 02 00 3D 00 17 08 F5 02 00 42 00 19 08
2860:F5 02 00 53 00 1B 08 F5 02 00 57 00 1D 08 F5 02
2870:00 5A 00 13 08 F5 02 00 5E 00 1B 08 F5 02 00 61
2880:00 1F 08 F5 02 00 64 00 15 08 F5 02 00 67 00 1D
2890:08 F5 02 00 6A 00 21 08 F5 02 F0 87 00 E9 03 F5
28A0:02 00 8A 00 E9 03 F5 02 00 B9 00 DF 07 F5 02 00
28B0:C0 00 0F 08 F5 02 00 C3 00 11 08 F5 02 00 C6 00
28C0:B2 07 F5 02 00 CF 00 B8 07 F5 02 00 D2 00 BA 07
28D0:F5 02 00 D5 00 BA 07 F5 02 00 D9 00 B8 07 F5 02
28E0:00 DD 00 C7 03 F5 02 00 E3 00 9A 03 F5 02 00 E9
28F0:00 9B 03 F5 02 00 F4 00 97 03 F5 02 00 F7 00 B0
2900:07 F5 02 00 0F 01 B0 07 F5 02 00 1C 01 97 03 F5
2910:02 00 27 01 97 03 F5 02 00 2B 01 0F 08 F5 02 00
2920:2E 01 0F 08 F5 02 00 35 01 97 03 F5 02 00 39 01
2930:11 08 F5 02 00 3C 01 11 08 F5 02 00 44 01 97 03
2940:F5 02 00 47 01 17 08 F5 02 00 4F 01 97 03 F5 02
2950:00 52 01 19 08 F5 02 00 5C 01 97 03 F5 02 00 60
2960:01 DF 07 F5 02 00 63 01 DF 07 F5 02 00 6A 01 97
2970:03 F5 02 00 6E 01 E1 07 F5 02 00 71 01 E1 07 F5
2980:02 00 77 01 97 03 F5 02 00 7E 01 97 03 F5 02 00
2990:83 01 9C 03 F5 02 00 88 01 9A 03 F5 02 00 8E 01
29A0:97 03 F5 02 00 99 01 97 03 F5 02 00 A4 01 DF 07
29B0:F5 02 00 AB 01 DF 07 F5 02 00 B0 01 E1 07 F5 02
29C0:00 B3 01 D7 01 F5 02 00 B6 01 89 03 F5 02 00 C3
29D0:01 B2 07 F5 02 00 C6 01 B2 07 F5 02 00 CE 01 B2
29E0:07 F5 02 00 D1 01 D7 01 F5 02 00 D8 01 B2 07 F5
29F0:02 00 E3 01 AC 03 F5 02 00 F0 01 13 08 F5 02 00
2A00:F3 01 C0 07 F5 02 00 F6 01 15 08 F5 02 00 F9 01
2A10:C2 07 F5 02 00 FC 01 9F 02 F5 02 00 09 02 1F 08
2A20:F5 02 00 0C 02 BC 07 F5 02 00 0F 02 21 08 F5 02
2A30:00 12 02 BE 07 F5 02 00 18 02 B4 07 F5 02 00 24
2A40:02 AC 03 F5 02 00 27 02 B4 07 F5 02 00 2D 02 DF
2A50:07 F5 02 00 30 02 C0 07 F5 02 00 33 02 E1 07 F5
2A60:02 00 36 02 C2 07 F5 02 00 39 02 9F 02 F5 02 00
2A70:3C 02 B4 07 F5 02 00 3F 02 B4 07 F5 02 00 48 02
2A80:13 08 F5 02 00 4B 02 BC 07 F5 02 00 4E 02 15 08
2A90:F5 02 00 51 02 BE 07 F5 02 00 57 02 AC 03 F5 02
2AA0:00 5A 02 1F 08 F5 02 00 5D 02 C0 07 F5 02 00 60
2AB0:02 21 08 F5 02 00 63 02 C2 07 F5 02 00 66 02 9F
2AC0:02 F5 02 00 6C 02 AC 03 F5 02 00 6F 02 1B 08 F5
2AD0:02 00 72 02 C0 07 F5 02 00 75 02 1D 08 F5 02 00
2AE0:78 02 C2 07 F5 02 00 7B 02 9F 02 F5 02 00 81 02
2AF0:AC 03 F5 02 00 84 02 DF 07 F5 02 00 87 02 BC 07
2B00:F5 02 00 8A 02 E1 07 F5 02 00 8D 02 BE 07 F5 02
2B10:00 90 02 0F 08 F5 02 00 93 02 C0 07 F5 02 00 96
2B20:02 11 08 F5 02 00 99 02 C2 07 F5 02 00 9C 02 9F
2B30:02 F5 02 00 A0 02 B2 07 F5 02 00 AC 02 C2 07 F5
2B40:02 00 B0 02 C0 07 F5 02 F0 B4 02 C4 07 F5 02 00
2B50:B7 02 C4 07 F5 02 F0 C7 02 C4 07 F5 02 00 CA 02
2B60:C4 07 F5 02 00 E1 02 C2 07 F5 02 00 E5 02 C0 07
2B70:F5 02 F0 F6 02 CD 07 F5 02 00 F9 02 CD 07 F5 02
2B80:F0 09 03 CD 07 F5 02 00 0C 03 CD 07 F5 02 00 25
2B90:03 C2 07 F5 02 00 29 03 C0 07 F5 02 00 3B 03 C0
2BA0:07 F5 02 00 3F 03 C2 07 F5 02 00 50 03 C2 07 F5
2BB0:02 00 54 03 C0 07 F5 02 00 58 03 BE 07 F5 02 00
2BC0:5C 03 BC 07 F5 02 F0 67 03 D6 07 F5 02 00 6A 03
2BD0:D6 07 F5 02 F0 7A 03 D6 07 F5 02 00 7D 03 D6 07
2BE0:F5 02 00 A0 03 B6 07 F5 02 00 AA 03 B6 07 00 00
2BF0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Une commande externe :

ShortP

G. Perreault

L'objet de la présente commande est de simplifier l'usage des préfixes ProDOS sous Basic.System. Le problème est le suivant :

Le programme Basic sur le disque dur se situe dans le dossier */Ddur20M/En.Cours/Compta/Version.2*, dossier qui contient le sous-catalogue *Donnees*. Pour travailler dans ce dernier dossier, il suffit de taper *Prefix donnees*, mais pour revenir au niveau précédent, il faudra taper *Prefix /Ddur20M/En.Cours/Compta/Version.2*...

Les stakhanovistes du clavier apprécient, les autres, dont je fais partie, taperont simplement `shortp`. Moins impressionnant mais plus élégant...

En pratique

Vous rendrez cette commande externe ProDOS disponible en tapant au clavier sous Basic :

`- shortp` ou `brun shortp`

Vous pourrez également l'activer depuis un programme Basic par :

```
10 print chr$(4) "- shortp
```

Dès lors pour raccourcir un préfixe, tapez au clavier : `shortp` , ou dans un programme Basic :

```
705 print chr$(4) "shortp
```

Pour contrôler l'effet de la commande, demandez à Basic.System le préfixe courant en tapant : `prefix`

Source Shortp.src
Assembleur ORCA/M

```
        err on
        65816 off
        65C02 off
        absaddr on
        keep shortp
        org $2000
main    START

; EQUATES:
```

```
; =======
; adresses en page zéro:
; ----------------------
length   gequ $2F            longueur de l'instruction
pcl      gequ $3A            adresse de la ligne désassemblée
a1       gequ $3C            adresse de départ pour move
a2       gequ $3E            adresse de fin     "     "
a4       gequ $42            adresse d'arrivée  "     "
adr      gequ $FC            pointeur temporaire
ptr      gequ $FE               "          "

; sous-programmes moniteur et adresses système:
; ---------------------------------------------
outdo    gequ $DB5C
insdsp2  gequ $F88C          désassemble la ligne (pcl)
move     gequ $FE2C          comme son nom l'indique

; ProDOS et BASIC.SYSTEM:
; -----------------------
mli      gequ $BF00          point d'entrée de ProDOS
extrncmd gequ $BE06          saut vers commande externe
errout   gequ $BE09          affiche erreur et fin
badcall  gequ $BE8B          convertit erreur MLI en BASIC.S
getbufr  gequ $BEF5          réserve (A) pages sous BASIC SYS.
xtrnaddr gequ $BE50          adresse commande externe
xlen     gequ $BE52          longueur de la commande
xcnum    gequ $BE53          n° de la commande
pbits    gequ $BE54          paramètres autorisés
vpathl   gequ $BE6C          vecteur vers la commande entrée
xreturn  gequ $BE9E          un RTS qui gagne à etre connu
opencnt  gequ $BE4D          nombre fichiers ouverts
long     gequ fin-debut+$100 longueur du programme à reloger

; ============================================================
;                          relogeur
; ============================================================
; Relogeur honteusement copié sur celui de Sylvie GALLET, qui
; a elle-meme copié celui de POM's n° 26

; initialisation
; --------------
init     lda mli
         cmp #$4C            est-ce un JMP ?
         beq init1           oui
         ldy nonactif        non -> affiche message et fin
nabcl    lda nonactif,y
         jsr outdo
         dey
         bne nabcl
         rts
init1    lda opencnt         y a-t-il des fichiers ouverts ?
         beq init2           non
         lda #$15            oui -> impossible placer la comma
                             nde message FILE(S) STILL OPEN
         jmp errout
init2    lda extrncmd+2      sauve adresse de la première
         sta precmd+2        commande externe
         lda extrncmd+1
         sta precmd+1
         lda #>long          demande d'un buffer de (A) pages
         jsr getbufr
         cmp #$0C            erreur ?
         bne gothen          non
         jmp errout          oui -> affiche 'NO BUFFERS AVAILA
                             BLE' et fin
gothen   sta adresse         n° de la 1-ère page accordée par
         sta a4+1            ProDOS
         sta extrncmd+2
         lda #>debut
         sta a1+1
         lda #<fin-1
         sta a2
         lda #>fin-1
         sta a2+1
         ldy #0
         sty a4              les adresses départ et arrivée
         sty a1              sont alignées -> poids faible = 0
         sty extrncmd+1
         jsr move            déplace le programme
         jsr reloge
         rts
reloge   lda adresse         adresse de relogement
         sta pcl+1              "
         lda #0                 "
         sta pcl                "
rdecode  ldx #0              nécessaire pour insdsp2
         lda (pcl,x)
         beq rfin            si 00 séparant progrmm et données
         jsr insdsp2
         ldy length          Y = longueur instruction - 1
         cpy #2              instruction sur 3 octets ?
         bne rsuivant        non -> pas de changement
         lda (pcl),y         oui
         cmp #>mli           est-ce un appel au MLI ?
         bne nomli           non
         dey                 peut-etre...
         lda (pcl),y
         cmp #<mli
         bne nomli           non
         dey
         lda (pcl),y
         cmp #$20            est-ce un jsr ?
         bne nomli           non
         lda pcl             oui -> avance pcl de 3 octets et
         clc                 traite les 3 octets suivant jsr $
                             BF00
         adc #3              comme une instruction à reloger
                             (l'adresse du buffer n'a pas beso
                             in d'etre relogée)
         sta pcl
         bcc nomli
         inc pcl+1
nomli    lda pcl             passage de paramètres pour calcul
         sta adr
         lda pcl+1
         sta adr+1
         ldy length
         jsr calcul
rsuivant lda length          on passe à instruction suivante:
         sec                 avec pcl = pcl + length
         adc pcl
         sta pcl
         lda pcl+1
         adc #0
         sta pcl+1
         sec                 distance (pcl) - adresse stockée
         sbc adresse         dans A (MSB) et X (LSB)
         ldx pcl
         jsr test            distance >= long du programme ?
         bcs rfin            oui -> on a fini
         jmp rdecode         non
rfin     rts
test     cmp #>fin-debut     A < nbre de pages du programme ?
         bcc oui             oui
         bne non             non et A > nbre de pages -> en de
                             hors du programme
         cpx #<fin-debut     on est dans la dernière page:
;                               X < poids faible de fin ?
         bcs non             non -> en dehors du programme
oui      clc                 dans les limites
         rts
non      sec
         rts
calcul   dey
         lda (adr),y         partie basse de l'opérande
         tax
         iny
         lda (adr),y         partie haute de l'opérande
         sec
         sbc #>debut         écart (en pages) entre adresse et
                             début
         bcc nonc            écart < 0 -> opérande en dehors d
                             es adresses à modifier
         jsr test            écart >= 0 -> écart > longueur du
;                                          programme ?
         bcs nonc                           non
         adc adresse                        oui -> on reloge
         sta (adr),y
         clc
         rts
nonc     sec
         rts
adresse  ds 1                adresse début du programme relogé
         ds $FF-(adresse-init)      debut sera égale à $2100

; ============================================================
```

```
;                    programme à reloger
; ==========================================================
debut    lda vpath1          vpath1 pointe sur la commande
         sta ptr             entrée
         lda vpath1+1
         sta ptr+1
         ldy #1
compare  lda (ptr),y         analyse de la commande tapée
         cmp commande-1,y
         bne nocmd           pas "shortp"
         iny
         cpy #7              longueur "shortp" + 1
         bcc compare
         lda (ptr),y
         cmp #$0D
         beq cmdok
         jsr $DAFB           envoie un retour chariot
         lda #$10            code de "SYNTAX ERROR"
         jmp errout
cmdok    dey
         dey
         sty xlen
         lda #0
         sta xcnum           n° commande externe
         sta pbits           aucun paramètre autorisé
         sta pbits+1           "       "       "
versuite lda xsuite          utile au relogeur seulement
         lda versuite+1      adresse de retour de ProDOS après
                              examen
         sta xtrnaddr        des paramètres
         lda versuite+2
         sta xtrnaddr+1
         clc                 retour à ProODS pour analyse des
                             paramètres
         rts                 suivant éventuellement la command
                             e tapée
nocmd    sec                 utile si notre cde est la seule
;                            ou la dernière pour signaler l'er
                             reur
precmd   jmp xreturn         saut à la commande externe
                             précédente
; Lecture du préfixe:
; ==================
xsuite   lda buffer          utile pour le relogeur (permet de
                              reloger l'adresse buffer utilisé
                              pour lire préfixe)
         lda xsuite+1
         sta table+1
         lda xsuite+2
         sta table+2
         jsr mli
         dc i1'$C7'          GETPREFIX
         dc i2'table'        table des paramètres utilisés pou
                             r GETPREFIX
         bcc ok
erreur   jsr badcall         affichage de l'erreur
         jmp errout            "       "
ok       ldy buffer          longueur du préfixe actuel
         beq end             si longueur = 0 inutile de raccou
                             rcir le
;                            préfixe
boucle   dey                 le dernier caractère est "/" donc
         lda buffer,y        inutile de le relire
         cmp #'/'
         bne boucle
         sty buffer          longueur du nouveau préfixe
         jsr mli
         dc i1'$C6'          SETPREFIX
         dc i2'table'        la meme table que pour GETPREFIX
         bcc end
         jmp erreur          en principe "PATH NOT FOUND"
end      rts
rien     ds 1                le relogeur l'apprécie
; Table de paramètres et message:
; ==============================
table    dc i1'1'            1 seul paramètre
         dc i2'buffer'       zone de stockage du préfixe
buffer   ds 64
commande dc c'SHORTP'
nonactif dc i1'25,13,13,7'
         dc c'** FITCA NON SODORP **'
fin      anop
         END
```

Récapitulation Shortp

Après avoir saisi ces codes sous moniteur, vous les sauvegarderez par :

```
bsave shortp, A$2000,L$1E8
```

```
2000:AD 00 BF C9 4C F0 0D AC
2008:CE 21 B9 CE 21 20 5C DB
2010:88 D0 F7 60 AD 4D BE F0
2018:05 A9 15 4C 09 BE AD 08
2020:BE 8D 4A 21 AD 07 BE 8D
2028:49 21 A9 01 20 F5 BE C9
2030:0C D0 03 4C 09 BE 8D E2
2038:20 85 43 8D 08 BE A9 21
2040:85 3D A9 E7 85 3E A9 21
2048:85 3F A0 00 84 42 84 3C
2050:8C 07 BE 20 2C FE 20 5A
2058:20 60 AD E2 20 85 3B A9
2060:00 85 3A A2 00 A1 3A F0
2068:50 20 8C F8 A4 2F C0 02
2070:D0 2C B1 3A C9 BF D0 19
2078:88 B1 3A C9 00 D0 12 88
2080:B1 3A C9 20 D0 0B A5 3A
2088:18 69 03 85 3A 90 02 E6
2090:3B A5 3A 85 FC A5 3B 85
2098:FD A4 2F 20 C8 20 A5 2F
20A0:38 65 3A 85 3A A5 3B 69
20A8:00 85 3B 38 ED E2 20 A6
20B0:3A 20 BA 20 B0 03 4C 63
20B8:20 60 C9 00 90 06 D0 06
20C0:E0 E8 B0 02 18 60 38 60
20C8:88 B1 FC AA C8 B1 FC 38
20D0:E9 21 90 0C 20 BA 20 B0
20D8:07 6D E2 20 91 FC 18 60
20E0:38 60 00 00 00 00 00 00
20E8:00 00 00 00 00 00 00 00
20F0:00 00 00 00 00 00 00 00
20F8:00 00 00 00 00 00 00 00
2100:AD 6C BE 85 FE AD 6D BE
2108:85 FF A0 01 B1 FE D9 C7
2110:21 D0 34 C8 C0 07 90 F4
2118:B1 FE C9 0D F0 08 20 FB
2120:DA A9 10 4C 09 BE 88 88
2128:8C 52 BE A9 00 8D 53 BE
2130:8D 54 BE 8D 55 BE AD 4B
2138:21 AD 37 21 8D 50 BE AD
2140:38 21 8D 51 BE 18 60 38
2148:4C 9E BE AD 88 21 AD 4C
2150:21 8D 86 21 AD 4D 21 8D
2158:87 21 20 00 BF C7 85 21
2160:90 06 20 8B BE 4C 09 BE
2168:AC 88 21 F0 16 88 B9 88
2170:21 C9 2F D0 F8 8C 88 21
2178:20 00 BF C6 85 21 90 03
2180:4C 62 21 60 00 01 88 21
2188:00 00 00 00 00 00 00 00
2190:00 00 00 00 00 00 00 00
2198:00 00 00 00 00 00 00 00
21A0:00 00 00 00 00 00 00 00
21A8:00 00 00 00 00 00 00 00
21B0:00 00 00 00 00 00 00 00
21B8:00 00 00 00 00 00 00 00
21C0:00 00 00 00 00 00 00 00
21C8:53 48 4F 52 54 50 19 0D
21D0:0D 07 2A 2A 20 46 49 54
21D8:43 41 20 4E 4F 4E 20 53
21E0:4F 44 4F 52 50 20 2A 2A
```

GS & FENETRES : Scrolling

Patrick Desnoues

Voici quelques considérations sur la mise à jour des fenêtres des programmes typiquement GS et une routine bien utile à l'écriture des applications.
À titre de démonstration, une application GS figure sur la disquette qui accompagne optionnellement ce numéro : elle vous permettra de lire les sources et fichiers de type TEXT.

Programmant depuis quelques temps sur Apple IIGS, je me suis heurté sur le fait de faire défiler du texte rapidement dans une fenêtre graphique. Beaucoup de programmes d'amateurs sont désespèrement lents, et je pensais cet état de chose normal, jusqu'au jour où j'ai vu ce que pouvaient faire les professionnels...

Je me suis donc penché sur le problème et après avoir décortiqué le déroulement des mises à jour, j'ai écrit cette routine qui pourra rendre service à un certain nombre de lecteurs programmeurs, où leur donner des idées pour faire mieux.

Défilement rapide de texte dans une fenêtre

La mise à jour d'une fenêtre (Update Event) survient dans plusieurs cas :

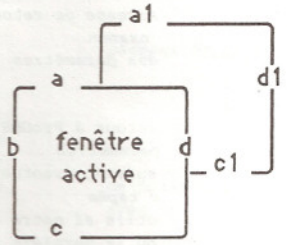
- clic dans une barre de défilement horizontal ou vertical ;
- changement de la taille d'une fenêtre (avec une exception) ;
- fermeture d'une fenêtre sur la fenêtre à mettre à jour (par exemple, fermeture d'un accessoire de bureau).

Une procédure de Quickdraw nous permet de connaître la taille du rectangle à mettre à jour, c'est `GetPortRect`.

Dans notre procédure, il faut donc :

1. Sauvegarder les valeurs du rectangle telles qu'elles nous avaient été retournés par `GetPortRect` la mise à jour précédente ;
2. Récupérer les nouvelles valeurs de `GetPortRect` ;
3. Faire une analyse des changements intervenus ;
4. Redessiner uniquement ce qui a pu être effacé.

La fenêtre n'est pas active



Il faut la redessiner de a1 à c1.

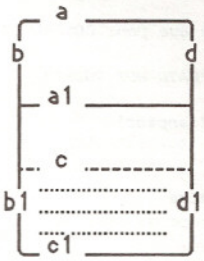
La fenêtre est active

➡ Si la taille de la fenêtre n'a pas changé

✗ C'est un clic sur ↓

a1 > a	b1 = b	c1 > c	d1 = d

- Le texte monte ;
- le Window manager déroule vers le haut ce qui est à l'intérieur.

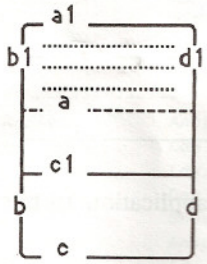


- a1 ≥ c : redessiner tout l'écran de a1 à c1 ;
- a1 < c : redessiner de c à c1.

✗ C'est un clic sur ↑

a1 < a	b1 = b	c1 < c	d1 = d

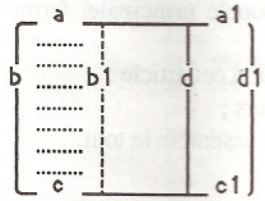
- Le texte descend ;
- le Window manager déroule vers le bas ce qui est à l'intérieur.



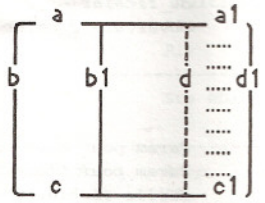
- c1 < a : redessiner tout l'écran de a1 à c1 ;
- c1 ≥ a : redessiner de a1 à a.

✗ C'est un clic sur ← ou →

a1 = a	b1 = b	c1 = c	d1 =



Flèche → :
- soit redessiner de a/b1 à b/c1 (le plus rapide) ;
- soit redessiner de a à c (le plus simple).

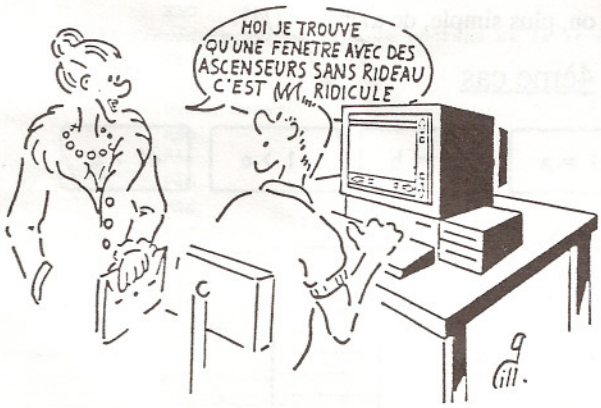


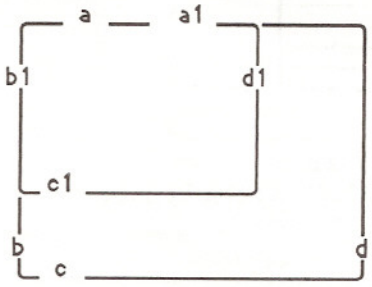
Flèche ← :
- soit redessiner de a1/d à d1/c1 (le plus rapide) ;
- soit redessiner de a à c (le plus simple).

➥ La fenêtre a changé de taille

✗ 1er cas

a1 = a	b1 = b	c1 < c	d1 < d

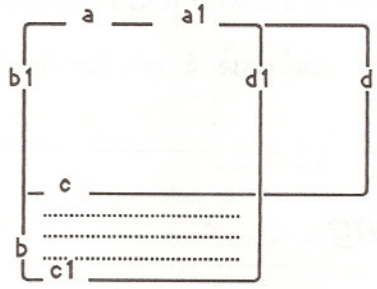




Le rétrécissement complet de la fenêtre est le seul cas qui n'entraîne pas de mise à jour. Ce qui est contenu dans la plus petite fenêtre tenait forcément dans la grande et n'a pas pu être effacé.

✗ 2ème cas

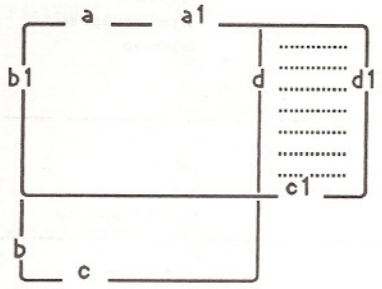
a1 = a	b1 = b	c1 > c	d1 < d



Il faut redessiner de c à c1.

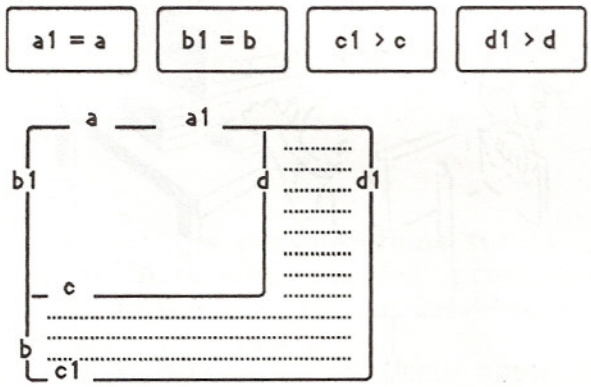
✗ 3ème cas

a1 = a	b1 = b	c1 < c	d1 > d



- Redessiner de a1/d à c1/d1 ;
- ou, plus simple, de a1/b1 à c1/d1.

✗ 4ème cas



La fenêtre a été agrandie. Il faut :
- soit redessiner de a1/d à c/d1 et de b/c à c1/d1 ;
- soit, plus simple, de a1/b1 à c1/d1.

Programmation

Il suffit de :
- créer une fenêtre avec des barres de défilement ;
- dans les paramètres de la fenêtre indiquer `'dc i4'Procedure'` dans le paramètre concerné ;
- stocker le numéro de la fenêtre retourné par `New Window` dans la variable `'AideWRefCon'` ;
- stocker l'adresse du texte à afficher dans la variable `'<Texte'` ;
- stocker la longueur du texte à afficher dans `'<LongTexte'` ;

C'est tout...

```
a = AncienHaut      a1 = NouveauHaut
b = AncienGauche    b1 = NouveauGauche
c = AncienBas       c1 = NouveauBas
d = AncienDroit     d1 = NouveauDroit
```

On peut encore optimiser la vitesse en calculant uniquement les mises à jour des scrolling horizontaux (comme cela est défini ci-dessus, mais non traité), mais pour l'affichage d'un texte standard, le jeu ne vaut guère le coût.

Une application

Sur le disque Pom's, vous trouverez l'application 16 bits 'Turbo.Scroll' à titre de démonstration.

Le programme a été écrit sous APW. Il permet de charger à partir d'un disque ProDOS des fichiers de type Text (T$04) ou Sources (T$B0). Après la transformation des minuscules accentuées adaptée à la langue française, le texte apparaît dans une fenêtre graphique avec une vitesse de défilement défiant pratiquement toute concurrence...

Les fichiers sur disque :

Turbo.Scroll	l'application GS ;
Librairie	équivalences et page zéro ;
Data	données et chaînes de caractères ;
MainPoms	initialisation, boucle principale, fermeture et Quit ;
Scrolling	tout ce qui à trait a cet article ;
FatalErreur	gestion des erreurs ;
LinkAll	fichier Exec qui assemble le tout.

Fichier Scrolling

```
Scrolling       START
                Using Ressource
                Using Datas
                Jsr SP_SFGetFile
                Jsr LoadFichier
                Rts
*---------------------------------------- STANDARD GETFILE
SP_SFGetFile    Anop
                Pushword #120           Position Horizont.
                Pushword #50                     Verticale
                Pushlong #D_SFGetFile   Titre
                Pushlong #0             Filtre
                Pushlong #NombreTC      Type accepté
                Pushlong #GoodSC        Réponse
                _SFGetFile
                Rts
*-------------------------------------------------------
LoadFichier     Anop
                Lda GoodSC              OK ?
                Bne OpenFichier
                Rts                     Non on annule
;-------------------------------------------------------
OpenFichier     Anop
                Pushzero Handle         Libère la place
                _DisposeHandle          du fichier
                _Open D_OpenParm        Ouverture fichier
                Bcc NoErr1
                Jmp ErreurOF
NoErr1          Anop
                Lda RefNum
                Sta RefNumRP            Param pour Read
                Sta RefNumCP            Param pour Close
                _Get_Eof RefNumCP       Taille fichier ?
                Bcs ErreurOF
                Lda LongueurOF
                Sta <LongTexte
                LongResult
                Pushlong LongueurOF     Taille à allouer
                Pushword MyID
                Pushword #%1100000000011000
                Pushlong #0
                _NewHandle
                Ply
                Plx
                Bcs ErreurOF
                Sty <Handle
                Stx <Handle+2
                Lda [Handle]            Où se trouve
                Tax                     le fichier ?
```

```
            Ldy #2
            Lda [Handle],Y
            Sta BufferData+2
            Sta <MyTexte+2
            Stx BufferData
            Stx <MyTexte
            _Read RefNumRP          Charge le fichier
            Bcs ErreurOF
            _Close RefNumCP
            Bcs ErreurOF
            Bra Analyse
ErreurOF    Anop
            Jsl SP_FatalErreur
            _Close RefNumCP         Ferme tout
            Rts
*---------------------------------------- Analyse de texte
Analyse     Anop
            Ldx #1                  Compte nb lig ($0D)
            Ldy Octets_Trans
            Dey
            Shortm
LoopCor     Anop
            Lda [MyTexte],Y
            Cmp #$0D
            Bne Lc0
            Inx                     Compte Nb ligne
DecremY     Anop
            Dey
            Bne LoopCor
            Bra StopAnalyse
Lc0         Anop
            Phx
            Ldx #CaractM-Caract0-1  Transforme les
                                    minusc accentuées
Lc1         Anop
            Cmp Caract0,X           Change les minusc
                                    anglaises -> VF
            Bne Lc2
            Lda CaractM,X
Lc1a        Anop
            Sta [MyTexte],Y         Stocke la valeur
            Bra Lc3                 corrigée
Lc2         Dex
            Bpl Lc1
Lc3         Anop
            Plx
            Bra DecremY
StopAnalyse Anop
            Longm
;------------------------ Calcul la taille de l'ascenseur
            LongResult
            Phx
            Pushword #$9            Hauteur Font = 9 *
            _Multiply               nb de ligne
            Pla
            Plx
;---------------------------------------- Ajuste l'ascenseur
            Sta WDataH              Nombre de lignes
                                    afficher
;------------------------------ Ouverture de la fenêtre
OpenWindow  Anop
            Pha
            Pha
            Pushlong #D_WindAide1
            _NewWindow
            PullLong WRefCon1
;------------------------------ Remet le rectangle à zéro
            Ldx #6
LoopRect1   Anop
            Stz NouveauRect,X
            Dex
            Dex
            Bpl LoopRect1
;------------------------------------------------------------
            Pushword #IFermer
            _EnableMItem
            Pushword #IOuvrir       Une seule fenêtre
            _DisableMItem           à la fois
            Rts
            END
*********** Procédure qui dessine le contenu de la fenêtre
Procedure1  START
            Using Ressource
            Using Datas
            Phb
            Phk
            Plb
            Phd
            Pha
            Lda RegistreD
            Tcd
            Pla
            Ldx #6
LoopRect    Anop
            Lda NouveauRect,X       Coordonnées de
            Sta <AncienRect,X       l'ancien rectangle
            Dex
            Dex
            Bpl LoopRect
            PushLong #NouveauRect   Coordonnées du
            _GetPortRect            nouveau rectangle
            Lda NouveauHaut
            Sta <PixelHaut
            Lda NouveauBas
            Sta <PixelBas
;------------------------------------------------------------
            Jsr CalcRegion
;------------------------------------------------------------
NonActive   Anop
            Stz <YPosition
            Ldy #-1
            Bra ChercDeb0
            Longa off
ChercheDebut Anop
            Lda [<MyTexte],Y
            Cmp #$0D
            Bne ChercDeb1
ChercDeb0   Anop
            Longm
            Clc
            Lda <YPosition
            Adc #Hauteur2Lignes     Ligne suivante +9
            Sta <YPosition
            Cmp <PixelHaut
            Bcs ChercDebFin
            Shortm
ChercDeb1   Anop
            Iny
            Cpy <LongTexte
            Bne ChercheDebut
            Longm
            Bra Sortie
;----------------------------
ChercDebFin Anop
            Iny
            Phy
            Pushword #10            Htab 30
            Pha                     = YPosition
            _MoveTo
            Ply
DebLigne    Anop
            Sty <PosDepart
            Stz <Compteur
            Shortm
FinLigne    Anop
            Lda [<MyTexte],Y
            Cmp #$0D
            Beq AfficheTexte
            Inc <Compteur
            Iny
            Cpy <LongTexte
            Bne FinLigne
            Dey
AfficheTexte Anop
            Phy
            Longm
```

Essai : ProCommand

Pierre Demblon

Ça y est, quelqu'un a craqué : à force que je répète tout le bien que je pensais de ce logiciel de Glen Bredon, et grâce à votre amicale insistance, on a trouvé à Pom's une courageuse pour traduire la documentation. Merci Joëlle. Dans ces conditions, j'aurais eu mauvaise grâce à refuser d'en faire la présentation.

D'autant que ni la documentation originale, ni même le nom du logiciel, ne permettent de voir globalement de quoi il s'agit. Glen Bredon le présente comme une suite de modules, de commandes hétéroclites qu'il a ajoutées peu-à-peu au Basic System de ProDOS en fonction de ses besoins. C'est beaucoup plus que cela : c'est l'Applesoft d'aujourd'hui.

Un Basic ancestral

Un peu d'histoire : l'Applesoft est le deuxième Basic qui ait existé sur l'Apple // (après l'héroïque Integer Basic légendairement écrit à la main, en hexa, par Wozniak). On l'appelait au début 'Floating Point Basic', parce qu'il était capable de diviser correctement. Il avait été écrit non par Apple, mais par Microsoft, en fonction des Basics comme des Apple existants à cette époque (fin des années 70) : commençant sur cassette, puis chargé d'une disquette, il a fini dans la Rom du tout nouvel Apple //, un ordinateur doué d'une capacité mémoire gigantesque pour l'époque : 64Ko ! Il permettait de gérer les caractéristiques extraordinaires de cet ordinateur cher mais magnifique, l'Apple II Plus : deux pages graphiques, en Haute Résolution en couleurs, par exemple.

Le problème, c'est que, pour l'essentiel, l'histoire de l'Applesoft s'est arrêtée là. Depuis, Apple s'est contenté de le reconduire dans la Rom de ses nouvelles machines, de l'interfacer avec les systèmes d'exploitation (DOS, ProDOS). Même les bugs sont restés, parce que les corriger aurait créé des incompatibilités avec les programmes existants, écrits en fonction d'eux.

Mais la vie a continué : le Basic a changé, l'Apple // aussi. Et les faiblesses de ce bon vieil Applesoft sont devenues de plus en plus criantes. S'il vit encore, c'est à cause d'une caractéristique typiquement Apple : l'ouverture. Je l'ai déjà écrit, les commandes les plus puissantes de l'Applesoft sont '&' et 'Call'. De plus, Apple a introduit dans Basic System, l'interpréteur Applesoft de ProDOS, une deuxième ouverture : la possibilité des commandes externes qu'on peut ajouter depuis le disque.

Et là, arrivent les utilisateurs, les bidouilleurs, les programmeurs, bref les Glen Bredon : et voilà pourquoi Applesoft est toujours vert. Autrement dit, si l'on veut comprendre ProCommand, il faut partir d'une liste des défauts de l'Applesoft : ProCommand, c'est la mise à jour permanente de l'Applesoft.

Par exemple : les capacités d'édition et de débuggage de l'Applesoft sont ridicules. Sa commande *List* est primitive. Son *Input* est plus que déficient. Son *Print* ne sait pas formater (*Print Using*). Il ne sait pas fusionner deux programmes ou changer globalement des numéros de ligne. Les conversions hexadécimal-décimal demandent une gymnastique fastidieuse. Il ne connaît ni la double haute résolution, ni la souris, ni la date et

```
                Lda <Compteur
                Beq LigneSuivante
                Pei <MyTexte+2
                Clc
                Lda <MyTexte
                Adc <PosDepart
                Pha
                Pei <Compteur
                _DrawText
LigneSuivante   Anop
                Pushword #10
                Clc
                Lda <YPosition
                Adc #Hauteur2Lignes
                Sta <YPosition
                Pha
                _MoveTo
                Ply
                Iny
                Cpy <LongTexte
                Beq Sortie
                Lda <PixelBas
                Adc #Hauteur2Lignes
                Cmp <YPosition
                Bcs DebLigne
Sortie          Anop
                Pld
                Plb
                Rtl
*********************************************************
CalcRegion      Anop
;-------------------------------------------------------
;  La fenêtre a-t-elle changée de taille ?
;-------------------------------------------------------
                Sec                     ; Test si modif
                                        horizontale
                Lda NouveauDroit        ; Si NouveauDroit
                                        - Anciendroit <> 0
                Sbc <AncienDroit        ; Oui : Changement
                                        de taille
                Bne ChangeTaille        ; Non : Test modif
                                        verticale
                Sec                     ; Test si modif
                                        verticale
                Lda NouveauGauche       ; Si NouveauGauche
                                        - AncienGauche <>0
                Sbc <AncienGauche       ; Oui : Changemen
                                        de taille
                Bne ChangeTaille        ; Aucune modif
;-------------------------------------------------------
; La fenêtre n'a pas changée de taille
;-------------------------------------------------------
FenetreIdem     Anop
                Lda <PixelHaut          ; Est-ce un scroll
                Cmp <AncienHaut         ing vers le bas ?
```

l'heure. Il n'est pas capable de faire la moitié de ce que ProDOS sait faire. J'arrête là, j'y serais encore demain.

D'autant que tout cela est réglé : avec ProCommand. ProCommand se présente effectivement comme un ensemble de 'modules' à ajouter 'à la demande' à Basic System. Mais ces modules ne sont pas là au hasard. Je vais donc tâcher de le présenter non pas module par module, mais par fonctions qui sont globalement ajoutées à l'Applesoft. Précisons seulement que tout cela se présente généralement comme des commandes de Basic System : les programmes Applesoft les traitent comme des commandes Basic ordinaires, on peut les taper en mode direct. Bref, l'Applesoft est étendu de trois ou quatre fois.

① Un véritable éditeur

☞ Edit

La commande *Edit* est un éditeur complet en une seule commande : un *Gple* complet en une commande qui occupe 1,5Ko. Déplacements du curseur, modes insertion et recouvrement, recherche et remplacement, compactage automatique des lignes, tout ce dont on peut rêver.

☞ Auto

La commande *Auto* permet évidemment, sans risques, la numérotation automatique des lignes.

☞ Macro

La commande *Macro* donne des macros clavier à la manière de *Gple* : ces macros sont bien entendu éditables.

☞ Lyst

La commande *Lyst* fait encore mieux que les macros de Dimitri Geystor pour AppleWorks (justification à droite des numéros de ligne par exemple) pour produire de superbes listings qui ridiculisent la commande *List*.

② Boîte à outils programmeur

☞ Renumber

La commande *Renumber* recalcule tous les numéros de ligne, vous donne accès aux numéros 'interdits' type 65531, respecte les programmes machine clandestinement ajoutés aux programmes Applesoft, marche avec les très gros programmes, accepte les '*List 10-20*' que les autres rejettent.

☞ Hold, Merge

Les commandes *Hold* et *Merge* permettent de fusionner deux programmes ou segments de programmes.

☞ Xref

La commande *Xref* affiche ou imprime les références croisées de toutes les variables du programme Applesoft en mémoire.

☞ Varlst, Vartrc

La commande *Varlst* affiche la liste des variables, la commande *Vartrc* permet de tracer vos variables bien sûr, mais admet également des expressions booléennes (elle vous permet par exemple de surveiller que la variable J reste inférieure à la variable I).

☞ Pointers

La commande *Pointers* vous permet de suivre en permanence les pointeurs de l'Applesoft (Himem etc).

☞ Hexdec

La commande *Hexdec*, vous vous en doutez, se charge des conversions hexadécimal-décimal.

Vous l'avez compris : adieu nos antiques Renumber du Dos Toolkit, adieu l'antique APA qui ne respectait pas les standards ProDOS. Mais aussi adieu GPLE. Même notre excellent EPE a trouvé un rude concurrent...

③ Les manques de l'Applesoft

☞ Using, Sci

Bien évidemment, Glen Bredon a comblé l'un des plus gros trous de l'Applesoft, l'absence de *Print Using*, avec ses commandes *Using* et *Sci* pour la notation scientifique des nombres. Mais les raffinements de cet *Using*-là, je ne les ai

```
                Beq Sal                 ; PixelHaut = Anci
                                        enHaut -> Scroll H
                Bcs ScrollBas           ; PixelHaut > Anci
                                        enHaut -> Scroll B
*                                       ; PixelHaut < Anci
                                        enHaut -> Scroll H
                Lda <PixelBas           ; Scroll Haut
                Cmp <AncienHaut         ; Petit scrolling?
                Bcc SortieAnalyse       ; Non
ScrollHautPix   Anop                    ; Oui
                Lda <AncienHaut         ; Calcul mise à jo
                                        ur minimum
                Sta <PixelBas
                Bra SortieAnalyse
ScrollBas       Anop                    ; Petit Scrolling
                                        vers le bas ?
                Cmp <AncienBas
                Bcc ScrollBasPix        ; Oui
                Rts                     ; Non
ScrollBasPix    Anop
                Lda <AncienBas          ; Calcul mise à
                Sta <PixelHaut          jour minimum
                Rts
SortieAnalyse   Anop                    ; Si on est en hau
                                        t de la fenêtre
                Lda <PixelHaut          ; inutile de calcu
                                        ler début Texte
Sal             Beq DebutConnu
                Rts
;-----------------------------------------------------------
; Si PixelHaut = 0). On ne cherche que la dernière ligne
; à mettre à jour car on sait où ça commence.
;-----------------------------------------------------------
DebutConnu      Anop
                Pla                     (Rts)
                Ldy #-1
                Lda #Hauteur2Lignes
                Sta <YPosition
                Jmp ChercDebFin
;-----------------------------------------------------------
ChangeTaille    Anop
                Lda <AncienDroit
                Cmp NouveauDroit
                Bcc PlusHaute
                Bra SortieAnalyse
;-----------------------------------------------------------
; La fenêtre est plus Haute
;-----------------------------------------------------------
PlusHaute       Anop
                Lda <AncienHaut
                Sta <PixelHaut
                Beq DebutConnu
                Rts
                END
```

pas vus sur beaucoup de nouveaux et 'SuperBasic'.

☞ Type

Type permet de lire et imprimer un fichier sans les défauts de l'*Input* de l'Applesoft, avec les paramètres de formatage que vous choisissez.

☞ Dump

Dump permet une lecture hexa/ASCII de la mémoire ou des fichiers ProDOS, avec paramètres toujours.

☞ Sort

Vous avez du mal à trier vos tableaux de variables, et ça prend un temps fou ? *Sort* est une commande magnifique : une base de données en une commande Basic ! Tri extrêmement rapide (Quicksort) alphabétique ou numérique d'un ou plusieurs tableaux de variables (chaînes, entiers, réels), à une ou à plusieurs dimensions, sur le champ voulu, sur un index du tableau, donc sans altérer le tableau. Vraiment magnifique.

☞ Rnd

Le générateur de nombres aléatoires de l'Applesoft est un peu lent, et il arrive souvent qu'il se répète ? Le *Rnd* de ProCommand ne se répétera pas avant 4 milliards de fois, et il est 15 fois plus rapide que celui de l'Applesoft.

☞ Input

Vous rêvez d'un *Input* qui accepte les ',', les ';', etc. ? Une mini-commande *Input* le fait.

☞ Rdline

Mais beaucoup mieux que ça : vous souhaitez une saisie avec des commandes d'édition sérieuses, à la AppleWorks ? Un input qui accepte juste les nombres ? Juste une date ? Juste une chaîne de 10 caractères ? Toujours en acceptant les virgules ? Qui convertit de minuscules en majuscules, ou le contraire ? Eh bien, c'est là environ 3% des possibilités de la commande *Rdline*, qui est un *Input* à rendre jaloux bien des applications : une sorte d'*'Input Using'* absolument magnifique. Entre *Sort* et *Rdline*, vous diminuez de moitié le temps d'écriture de certains programmes. À elles seules, ces deux commandes valent les 500,00 francs.

④ L'interpréteur ProDOS

☞ Copy

Basic System, vous le savez, ne donne pas accès à certaines possibilités de ProDOS, pour lesquelles nous devions jusqu'ici passer par le langage-machine. Regardez par exemple les contorsions que doivent employer les Beagle pour copier, par l'Applesoft, les fichiers TimeOut sur votre disquette AppleWorks : ils n'ont pas de commande *Copy*.
Bien entendu, cette commande *Copy* qui copie les fichiers d'un catalogue à l'autre est dans ProCommand.

☞ Online

De même qu'une commande *Online* qui regarde quels sont les volumes ProDOS présents et donne leurs noms.

☞ Date, Datestr

ProDOS connaît la date et l'heure, si vous avez une horloge, mais pas Basic System. Les commandes *Date* et *Datestr* vont s'en charger, en format 12 heures ou en format 24 heures, et vous donneront même le jour de la semaine. (Note : je parie que nous n'allons pas attendre longtemps pour qu'un lecteur de Pom's francise les noms de mois et de jours dans cette commande).

☞ Clock

Vous n'avez pas d'horloge ? Eh bien, si vous avez un //c, ProCommand vous en donne une, en utilisant les interruptions de la souris, avec la commande *Clock*.

☞ Pop

Les chemins d'accès du ProDOS vous embêtent ? Je ne sais pas comment résoudre cette question en théorie : comme fanatique de ProDOS, vous me connaissez, je suis aussi un fanatique des noms de volume et chemins d'accès. Mais c'est vrai que taper 64 caractères sans faute, ce n'est pas la joie tous les jours. En théorie, je ne sais pas. Mais en pratique, la réponse est ProCommand :
Par exemple, quand vous êtes enfoncé dans un préfixe dément du genre /hard/boulot/AppleWorks/data/poms/P44/ procmd, la commande *Pop* vous permet de remonter d'un niveau de préfixe.

☞ Path

La commande *Path* permet de retrouver automatiquement un fichier dans n'importe quel sous-catalogue d'un volume donné, et fixe automatiquement le préfixe sur son chemin d'accès.

☞ Find

Une fois que vous avez retrouvé le fichier, la commande *Find* va vous permettre de retrouver dans ce fichier n'importe quelle chaîne recherchée, en ASCII ou en hexa, en vous donnant le décalage depuis le début du fichier.

☞ Compare

Vous pouvez même comparer deux fichiers pour voir quelles modifications ont été introduites (quels patches du patcheur fou vous avez mis dans AppleWorks par exemple), avec la commande *compare*. Ce qui est très précieux avec cette commande, c'est que vous pouvez demander une comparaison –avec décalage– : si un paragraphe a été ajouté dans un fichier texte, ou qu'un fichier en langage-machine a été réassemblé, vous pouvez avoir une comparaison de ce qui est comparable.

☞ Paste

Vous pouvez bien entendu coller un deuxième fichier à la fin du premier avec la commande *Paste*.

☞ Setinfo

Basic System ne nous donne qu'un accès partiel aux attributs d'un fichier dans les catalogues ProDOS. ProCommand donne accès à chaque attribut. Par exemple l'octet des paramètres d'accès, qui se résume sous Basic System à *Lock* et *Unlock*, permet sous ProDOS des choses bien plus amusantes : autoriser ou interdire séparément la lecture ou l'écriture d'un fichier, sa destruction, la possibilité de le renommer. Il a un bit spécialisé qui permet de savoir si le fichier a déjà ou non été recopié (très pratique pour les 'backups' sélectifs). Avec la commande *Setinfo* de ProCommand, vous pouvez jouer à autoriser l'écriture d'un fichier, mais pas sa lecture, et autres fantaisies.
Mais elle permet également de changer le type d'un fichier, ce qui est très précieux, pour les polices de Publish-It par exemple. Surtout, elle permet enfin l'accès à l'octet du type auxiliaire, qui devient de plus en plus important sous ProDOS 8 et 16. (Pour la lecture de cet octet sous Basic System, utilisez le patch de J.Y. Bourdin publié dans Pom's 36 page 62).

☞ Eject, Format

Enfin, ProDOS 8 lui-même, confiné qu'il est dans les 16Ko de la carte langage, ne peut pas tout faire : pour formater un disque, ou éjecter une disquette 3,5", il faut passer par un programme-machine. Qu'à cela ne tienne, répond Glen Bredon, il suffit de présenter les programmes machine en question sous forme de commande externe, et Basic System saura le faire.

La commande *Eject* se charge donc du grille-pain, tandis que la commande *Format* a été faite, souvenez-vous, pour faire plaisir à Pierre Demblon.

⑤ Double Haute Résolution

☞ DHGR

L'Applesoft a, à peu près, tout ce qu'il faut pour gérer la Haute Résolution : *Hplot* et *Hcolor*, *Draw* et *Xdraw*. Mais pour la double haute résolution, bernique. ProCommand se charge bien entendu, avec des commandes '&' cette fois, de nous fournir l'équivalent pour la Double Haute Résolution des commandes HGR, en 16 couleurs. Les programmes de démo de la disquette sont extrêmement impressionnants !

☞ Fill

Un module *Fill* vous permet de remplir très rapidement en couleurs, par une simple commande en Basic, des figures tracées. Une palette de différents mélanges de couleurs est permise, avec un nombre de combinaisons quasi-infini. Là aussi, faites tourner les démos.

☞ Hgr

Je suis plutôt impressionné, certains jours, par des programmes 8 bits 128Ko dont l'affichage se fait en double haute résolution noir et blanc, comme Publish-it ou Springboard Publisher, et je me demande à quoi sert, en monochrome, la super haute résolution du GS. L'explication est simple : pour le moment (mais ça changera un jour), la résolution maximale du GS est de 600 points horizontaux sur 200 points verticaux. Or, en double haute résolution monochrome, on atteint 560 points sur 192 : la différence est minime.
Oui, mais comment faire, vous et moi, pour faire aussi bien que Publish-it ? Le module HGR de ProCommand, bien sûr.

☞ Rgb

Bien entendu, ça ne vous suffit pas, vous voulez mélanger les deux modes couleur et monochrome, 'à la Arlequin'. Bien entendu, Glen Bredon aussi. Puisqu'il faut pour cela une carte couleur RVB type Féline ou un GS, Bredon a mis un module RGB qui s'en charge, toujours sous Basic System.

☞ Print Hgr

Ça ne vous suffit toujours pas, vous voulez un générateur de caractères en double haute résolution, faire *Print* sur l'écran graphique DHGR ? Le module y est, bien sûr. Vous voulez profiter de ce *Print* pour faire des animations DHGR ? Il le fait, bien sûr.

☞ Éditeur de polices

Vous voulez un éditeur pour ces polices graphiques et ces shapes ? Il y est, bien sûr.

☞ Hload, Hsave

Vous souhaitez, indépendamment de tout module graphique, pouvoir charger, voir à l'écran, et sauver, les images DHGR type Dazzle Draw ? Les commandes *Hload* et *Hsave* s'en chargent. Pratique, pour les Slide Show. Vous trouvez que ces images Dazzle Draw sont bien grosses avec leurs 33 blocs : les mêmes commandes se chargent, si vous le souhaitez, de compresser et décompresser les fichiers images !

Bref, comme d'habitude avec Bredon, il vaudrait mieux faire la liste de ce que son programme ne fait pas, que la liste de ce qu'il fait : ce serait plus court...

⑥ La souris

Essayez donc, sous Applesoft, de déplacer une 'shape' (un curseur par exemple), avec la souris. Ou bien de lire la position de la souris :
amusez-vous bien ! La documentation donnée par Apple avec la souris //e dit ce qu'il faut faire, mais quelle galère...

☞ Mouse

Jusqu'ici, j'utilisais un excellent programme, méconnu et c'est bien dommage, publié par Apple lui-même : il s'appelle Apple II DeskTop Toolkit consiste en 4 disquettes 140Ko, 450 pages de documentation, et fournit l'ensemble des routines nécessaires pour interfacer vos programmes avec la souris, tant en mode texte qu'en mode graphique, tant en assembleur qu'en Pascal ou qu'en Basic.

Ce programme était destiné aux développeurs (mais nous sommes tous des développeurs). Si vous ne l'avez pas, dépêchez-vous : l'APDA le diffuse encore, mais dans sa boutique des 'antiquités et curiosités' (il date de 85, les choses vont vite ces temps-ci). En fait, vous le connaissez : nombre d'applications se résument pour l'essentiel à de la (bonne) sauce mise autour de ces routines (l'excellent MouseWrite, chez Roger Wagner, par exemple). Nous faisons du Desktop Toolkit comme Mr Jourdain faisait de la prose.

L'avantage du module *Mouse* de ProCommand, c'est qu'il est parfaitement intégré avec ses autres routines de gestion de la Double Haute Résolution. Faites tourner le programme de démo (Demo Mouse), et vous m'en direz des nouvelles.

☞ Reset

La mémoire de l'Apple // n'étant pas infinie, il faut gérer tout ça ensemble. Il va bien falloir un jour enlever les commandes supplémentaires pour laisser la place au programme lui-même, par exemple. Comment faire ? Eh bien, c'est très simple : utilisez la commande *Reset*, de ProCommand, bien entendu !

☞ Cmds

ProCommand semble avoir toujours la réponse. Mais sur là, il ne m'aura pas. C'est qu'il y a une limite à ce système des commandes à Basic System du ProDOS, précisément ce fait qu'on ne peut pas les utiliser toutes à la fois, puisqu'il faut rajouter chaque commande en mémoire. Or la solution est connue, elle est dans des interpréteurs comme Davex, ou dans des systèmes d'exploitation dont je n'ose pas écrire le nom ici : *les commandes non-résidentes*. Vous tapez la commande avec ses paramètres, et si le système d'exploitation ne l'a pas en mémoire, il va la chercher sur le disque, l'éxécute, puis libère la mémoire qu'elle a occupée. Apple n'a pas pensé à ça, dans Basic System. Ah, coincé, ce coup-là, n'est-ce pas ?

Vous vous en doutez : encore perdu, Demblon ! Ce damné Glen Bredon a mis une commande *Cmds* qui va chercher les commandes sur le disque : avec la commande *Reset*, ça règle la question. Et inutile de dire «ça va planter si tu changes de disquette», parce que, bien sûr, la commande *Cmds* permet de choisir le chemin d'accès ProDOS des commandes additionnelles.

Et si le type CMD que Glen Bredon donne à ses commandes ne vous convient pas (avec le type BIN on peut faire un 'brun' ou un '–'), eh bien utilisez donc la commande Setinfo... de Glen Bredon.

Je l'ai déjà écrit : cet homme est un génie. Chapeau.

C'est une conclusion suffisante, je pense.

Graphisme & Son :

DPA 23,24,25 & 26

Nos quatre nouvelles DPA de ce numéro ont une orientation totalement *Graphiques* et *Sons*. Elles ne sont disponibles évidemment qu'en format 3,5" pour le GS.

Nous avons poursuivi le travail de classement et de sélection de ces images et sons du domaine public. Vous trouverez l'essentiel de ce qu'il y a à savoir sur ceux-ci dans les numéros 42 et 43 de Pom's, et dans le fichier *Lisez-moi* sur les disquettes.

DPA.023 et DPA.024
Images GS 3 et 4

Rappelons seulement quelques choix que nous avons faits :

- Nous avons choisi le format de fichier, pourtant obsolète, de GS Paint ($C0, type auxiliaire 0). Et cela, tout d'abord parce que GS Paint, donné avec le GS, et qui est une bonne application, est entre les mains de tous les GSphiles français. De plus, ce format est le plus compressé, et permet de mettre le maximum d'images sur chaque disque. Enfin, il permet d'obtenir des images de deux écrans de haut, et de les imprimer. Ce format est accepté tel quel par AppleWorks GS et PaintWorks Gold. Pour passer ces images dans d'autres applications, le mieux est de les passer d'abord dans GS Paint pour convertir en fichiers écran de type $C1.

- À noter que si le nom du fichier a le suffixe ".640", cela signifie qu'il est en mode 640, avec *Dithering*. S'il a le suffixe ".DBL", cela signifie que c'est un dessin de deux écrans, une page imprimée.

- Vous trouverez sur ces disquettes une application de bureau freeware appelée **Panoramix** (à lancer sous GS/OS) qui permet de faire défiler les images de toutes les façons possibles, et d'intervenir sur les fichiers. Faites Ó-L pour voir l'ensemble des images d'un disque.

- Que faire avec toutes ces images ? Il y a des réponses évidentes : les regarder, tout d'abord (certaines sont magnifiques, et feraient à elles seules vendre des GS). Mais aussi, bien entendu, jouer au livre d'images, au jeu du coloriage sur GS : quand on voit ce qu'on peut faire avec GS Paint et ces images, on sourit de nos antiques "coloriages" VersionSoft. Donnez ces images et GS Paint à un enfant, et il n'est pas près de vous rendre votre GS. Et n'oubliez pas les animations GS Paint !

On peut aussi bien entendu les imprimer (en couleurs éventuellement). Elles rentrent donc comme composant dans notre PAO personnelle, nos en-têtes de lettres, cartes de vœux, etc.
Mais nous voudrions vous suggérer une autre utilisation particulièrement amusante : les puzzles. Avec le programme de **Puzzle** de **Pom's 43**, vous créerez des puzzles très amusants, et certains seront atrocement difficiles. Mais là, au moins, vous ne perdrez pas de pièces, et vous pourrez interrompre et reprendre quand vous voudrez. Attention : il faut passer ces dessins dans GS Paint et les sauver en format écran 65 blocs, les décompresser donc, pour en faire des puzzles.

La disquette DPA.023...

...comporte 45 images. Tout d'abord, une série d'agréables paysages, généralement paisibles et reposants : le paysage permet un jeu des couleurs et des dégradés où le GS révèle ses capacités.

Nous avons équilibré cet apaisement avec une série d'images d'horreur (images d'horreur ne signifiant pas images horribles !), dans un sous-catalogue appelé

/Frissons. Quelques vampires et autres étranges visiteurs pour peupler nos cauchemars.

Pour retourner à la beauté (!), mais aussi un peu à l'humour, reconnaissons-le, nous avons mis un sous-catalogue /REDACTION.POMS dans lequel vous trouverez... nos portraits. Oui, nous sommes tous là : Ariel Sebban, Christian Piard, Daniel Lurot, Éric Weyland, Hervé Thiriez, Jean-Luc Bazanègue, Jean-Yves Bourdin et Joëlle Piard. Certains d'entre nous ont même deux portraits, et un portrait collectif du conseil de rédaction couronne le tout.

Nous ne vous cacherons pas que nous nous sommes félicités d'avoir décidé, *à l'avance*, que nul n'aurait le droit de retoucher son portrait ou de refuser le travail des artistes...

Vous trouverez enfin sur cette disquette 14 images de Science-Fiction dans le sous-catalogue /SF. Après le réalisme des portraits, le déchaînement de l'imagination des artistes...

La disquette DPA.024...

...comporte 50 images, et elle est plus orientée vers l'art : qu'il s'agisse de la peinture ou du cinéma, ou de la littérature, avec des images d'aventures, de chevaliers, de pirates, ou d'études proprement artistiques dont certaines (Africa, Age de Pierre, etc., sont à notre avis vraiment belles), ou de la musique.

Nous avons donc complété ces images avec des images plus rudes : une série de guerriers, du sport, et avec quelques images humoristiques qui nous ont vraiment amusés (par exemple le "Hit Any Key to continue...").

DPA 025 et 026

Sons GS Volume 3 et 4

Deux nouvelles disquettes pleines de sons numérisés (tous différents, même si les noms sont voisins). Rappelons que si une carte de numérisation est nécessaire pour enregistrer ces sons, il n'est besoin d'aucune carte particulière pour les éditer, les mettre dans vos programmes, etc., une fois qu'ils sont sur la disquette. C'est la même chose qu'avec les images numérisées.

Répondant à votre demande, nous avons cette fois-ci mis également des musiques enregistrées, et des sons plus longs, quitte à en mettre moins. Nous avons également recherché la qualité de la reproduction avec des sons à échantillonnage plus fréquent, qui sont donc à jouer plus rapidement.

Comment gérer tous ces sons ? On peut tout d'abord les éditer, pour les travailler, les couper-coller, les ralentir ou les accélérer, leur donner de l'écho, etc. C'est l'objet de l'excellente application freeware **SoundStudio** que vous trouverez sur chaque disquette de sons (à lancer sous GS/OS). Pom's recommande aussi l'application **Sound Shop**, incluse dans HyperStudio.

Pour inclure ces sons dans vos programmes, nous vous recommandons bien entendu HyperStudio : nous sommes dans l'ère de l'Hypermédia, nos programmes doivent désormais être "graphiques/sons", et HyperStudio, comme Hypercard sur le Macintosh, est ce qui le permet le plus facilement.

Mais n'oubliez pas trois autres usages de ces sons, usages amusants et faciles :

- travailler sous AppleWorks GS (ou autre) avec une musique de fond, qui joue en même temps que vous travaillez (eh oui, le GS le permet): c'est possible avec le NDA **BackGround Music** de la disquette DPA 11.
- lancer un de ces sons au démarrage de la disquette : c'est possible avec l'Init (ou TIF : les Init du GS sont des Pif et des Tif) **StartSound** de la DPA 20.
- remplacer le Bip du GS par un son de votre choix (voyez le passage sur *AppleWorks-Tarzan* dans Apple // For Ever du numéro 43). C'est ce que permet l'Init (PIF cette fois-ci) **Sysbeep** que vous trouverez sur la DPA 10. Essayez plusieurs sons : à l'usage, il s'avère que si certains sons sont lassants comme messages d'erreur, d'autres continuent à faire rire à chaque fois.

Trois précisions enfin:

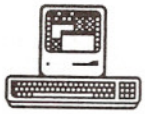
- Le suffixe ".200", ".400", etc., dans le nom du fichier indique la vitesse à laquelle, à notre avis, ce son doit être entendu. Mais bien entendu, une des premières choses à faire avec SoundStudio, c'est de faire varier cette vitesse, pour s'amuser. Cette valeur est déposée dans le type auxiliaire du fichier.
- Beaucoup de ces sons, en particulier les plus courts, doivent être entendus en répétition permanente (icône circulaire dans SoundStudio), par exemple les bruits de machine à écrire ou de mitraillette.
- N'oubliez pas que la sortie son du GS accepte un casque stéréo : la famille n'apprécie pas toujours quand vous bricolez vos sons sur le GS...

Liste des sons de la DPA 025

BEETHOVEN (une très belle musique, qui occupe la moitié de la disquette). BOUM. BRUITAGE. CHIENS. DINDES. ÉLÉPHANT. GOOD.JOB. INCORRECT. LION. MOUU. RIGHT. RIRE.1. RIRE.2. RIRE.3. RYTHME.

Liste des sons de la DPA 026

APPLAUDISSEMENTS. BRIS. CARILLON. CLOCHES. CLOCHETTE. DONG1. DONG2. EAU. EUUAA. EXPLOSION. HARMONIE. LASER1. LASER2. MACHINE. ÉCR. NAGE. SAUT. SIFFLET. SPLASH. TINTE1. TINTE2. TINTE3. TIRS. ABCD. WARNING. A.1.2.3.4. BANJO. BOING. BONG. CARILLON. CORNE. DONG. RYTHME. SHIT. TIR. VERY.GOOD. WOOPS. DISCO. TROMPETTES

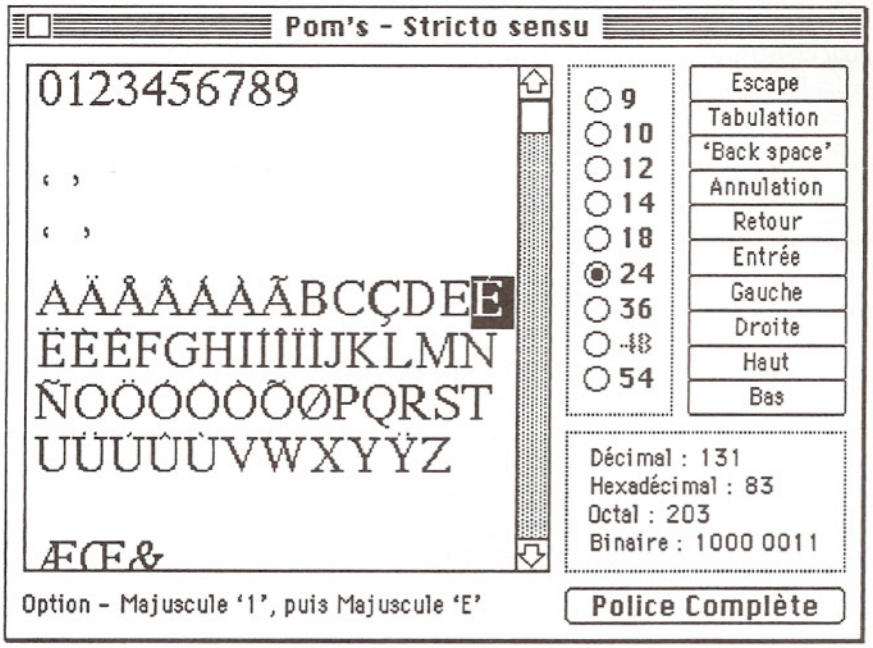


Un accessoire Mac :

Stricto Sensu

Jean-Luc Bazanegue

Pom's a déjà publié deux accessoires de bureau destinés à la recherche des caractères dans les différentes polices : *Lucy* et *ad litteram*. Le second est devenu depuis un standard, mais il lui manquait cependant quelque chose : l'affichage de la séquence de touches nécessaire à l'entrée au clavier de caractères 'exotiques'. Par exemple, pouvez-vous nous dire, comme ça, sans réfléchir, comment on obtient le caractère 'É' ? Pas évident, hein ! Et bien, en faisant Option - Majuscule 1, puis Majuscule E, et cela Stricto sensu peut vous l'indiquer.



Accessoirement, et surtout pour les développeurs, *Stricto sensu* affiche le code décimal, hexadécimal, octal et binaire des caractères ou des touches spéciales du Macintosh.

Utiliser Stricto sensu

Choix de la police de caractères

Lors de l'ouverture de l'accessoire, la police affichée par défaut est 'Times' si elle est présente dans le système ; c'est la plus complète.

Sinon, *Stricto sensu* utilise 'Chicago' qui est toujours présente puisque faisant partie du 'système Macintosh'. Pour changer la police de caractères, utilisez le menu 'Police' qui s'installe dans la barre de menu au moment de l'ouverture de l'accessoire. Il donne accès à toutes les polices présentes.

Choix de la taille des caractères

À l'ouverture de *Stricto sensu* —comme au moment d'un changement de police—, une routine vérifie si la police existe en 24 points. Si c'est le cas, cette taille est choisie. Sinon, la routine continue ses recherches en allant vers 9 points (24, 18, 14, 12, 10 puis 9 points). La première taille valide trouvée est employée. Dans le cas où aucune de ces tailles n'existerait —c'est rare— le processus repart dans l'autre sens, de 36 à 54 points.

Pour un changement 'manuel' de la taille des caractères, utilisez les boutons de taille situés à droite du texte. Seules les tailles valides sont proposées.

Séquence de touches et codes des caractères

La sélection du caractère pour l'affichage de la séquence de touches et des codes du caractère peut se faire de trois manières différentes :

Barre d'insertion :
l'affichage concerne le caractère qui précède la barre d'insertion. Ceci est pratique lorsque l'on veut obtenir le code d'un caractère frappé directement au clavier.

Un caractère sélectionné :
l'affichage concerne naturellement le caractère sélectionné.

Plusieurs caractères sélectionnés :
l'affichage concerne le premier caractère de la sélection.

Il est aussi possible de se déplacer dans le texte à l'aide des touches fléchées pour afficher les informations sur les caractères.

Touches 'spéciales'

Pour obtenir les codes d'une des touches spéciales du Macintosh, il suffit de cliquer sur un des boutons situés en haut et à droite de la fenêtre.

Bouton 'Police complète'

Il permet d'obtenir à nouveau la liste complète des caractères —classés par 'thèmes'— après une édition.

Notes

Lorsque l'accessoire affiche "Impossible : utilisez le 'Copier/Coller'", cela veut dire que nous n'avons pas trouvé la méthode pour obtenir au clavier le caractère considéré ; il est néanmoins possible que certaines séquences de touches particulièrement 'tordues' nous aient échappé. Dans ce cas, faites nous part de vos découvertes et nous modifierons l'accessoire en conséquence.

Stricto sensu *fonctionne sur tous les Macintosh. En revanche, l'affichage des séquences de touches ne sera exact que si les trois éléments suivants sont réunis :*

- *Macintosh plus, SE, II... Le clavier du Mac 512 ne convient pas ;*
- *clavier 'AZERTY' ;*
- *Système français (ou francisé).*

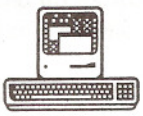
```
#include <MacTypes.h>
#include <DeviceMgr.h>
#include <WindowMgr.h>
#include <MenuMgr.h>
#include <EventMgr.h>
#include <ToolboxUtil.h>
#include <QuickDraw.h>
#include <TextEdit.h>
#include <pascal.h>

/*********************************************/
#define  Oui          1
#define  Non          0

#define  t9           0
#define  t10          1
#define  t12          2
#define  t14          3
#define  t18          4
#define  t24          5
#define  t36          6
#define  t48          7
#define  t54          8
#define  nScroll      9
#define  pComp        10

#define     Impossible  0
#define     Une_fois    1
#define     Deux_fois   2
#define     Repetition  3

#define     Touche      0
#define     Majusc      1
#define     Maj         1
#define     Option      2
#define     Op          2

/*********************************************/
Byte    st[1380] =
   {
    48,Une_fois ,Touche, 48,0,0,                    /* 0*/
    49,Une_fois ,Touche, 49,0,0,                    /* 1*/
    50,Une_fois ,Touche, 50,0,0,                    /* 2*/
    51,Une_fois ,Touche, 51,0,0,                    /* 3*/
    52,Une_fois ,Touche, 52,0,0,                    /* 4*/
    53,Une_fois ,Touche, 53,0,0,                    /* 5*/
    54,Une_fois ,Touche, 54,0,0,                    /* 6*/
    55,Une_fois ,Touche, 55,0,0,                    /* 7*/
    56,Une_fois ,Touche, 56,0,0,                    /* 8*/
    57,Une_fois ,Touche, 57,0,0,                    /* 9*/
    32,Une_fois ,Touche, 32,0,0,                    /*  */
    32,Une_fois ,Majusc, 32,0,0,                    /*  */
    65,Une_fois ,Majusc, 65,0,0,                    /* A*/
   128,Impossible,Touche,128,0,0,                   /* Ä*/
   129,Une_fois ,Maj+Op, 90,0,0,                    /* Å*/
   229,Une_fois ,Option,122,0,0,                    /* Å*/
   231,Impossible,Touche,231,0,0,                   /* Á*/
   203,Deux_fois ,Touche, 96,Majusc, 65,            /* À*/
   204,Deux_fois ,Option,110,Majusc, 65,            /* Ã*/
    66,Une_fois ,Majusc, 66,0,0,                    /* B*/
    67,Une_fois ,Majusc, 67,0,0,                    /* C*/
   130,Une_fois ,Option,141,0,0,                    /* Ç*/
    68,Une_fois ,Majusc, 68,0,0,                    /* D*/
    69,Une_fois ,Majusc, 69,0,0,                    /* E*/
   131,Deux_fois ,Maj+Op, 49,Majusc, 69,            /* É*/
   232,Impossible,Touche,232,0,0,                   /* Ê*/
   233,Impossible,Touche,233,0,0,                   /* È*/
   230,Impossible,Touche,230,0,0,                   /* Ê*/
    70,Une_fois ,Majusc, 70,0,0,                    /* F*/
    71,Une_fois ,Majusc, 71,0,0,                    /* G*/
    72,Une_fois ,Majusc, 72,0,0,                    /* H*/
    73,Une_fois ,Majusc, 73,0,0,                    /* I*/
   234,Impossible,Touche,230,0,0,                   /* Î*/
   235,Impossible,Touche,230,0,0,                   /* Î*/
   236,Impossible,Touche,230,0,0,                   /* Ï*/
   237,Impossible,Touche,230,0,0,                   /* Î*/
    74,Une_fois ,Majusc, 74,0,0,                    /* J*/
    75,Une_fois ,Majusc, 75,0,0,                    /* K*/
    76,Une_fois ,Majusc, 76,0,0,                    /* L*/
    77,Une_fois ,Majusc, 77,0,0,                    /* M*/
    78,Une_fois ,Majusc, 78,0,0,                    /* N*/
   132,Deux_fois ,Option,110,Majusc, 78,            /* Ñ*/
    79,Une_fois ,Majusc, 79,0,0,                    /* O*/
   133,Deux_fois ,Majusc,172,Majusc, 79,            /* Ö*/
   238,Impossible,Touche,238,0,0,                   /* Ô*/
   239,Impossible,Touche,239,0,0,                   /* Ô*/
   241,Impossible,Touche,241,0,0,                   /* Ò*/
   205,Deux_fois ,Option,110,Majusc, 79,            /* Ô*/
   175,Une_fois ,Maj+Op, 48,0,0,                    /* Ø*/
    80,Une_fois ,Majusc, 80,0,0,                    /* P*/
    81,Une_fois ,Majusc, 81,0,0,                    /* Q*/
    82,Une_fois ,Majusc, 82,0,0,                    /* R*/
    83,Une_fois ,Majusc, 83,0,0,                    /* S*/
    84,Une_fois ,Majusc, 84,0,0,                    /* T*/
    85,Une_fois ,Majusc, 85,0,0,                    /* U*/
   134,Deux_fois ,Majusc,172,Majusc, 85,            /* Ü*/
   242,Impossible,Touche,242,0,0,                   /* Ú*/
   243,Impossible,Touche,243,0,0,                   /* Û*/
   244,Impossible,Touche,244,0,0,                   /* Ù*/
    86,Une_fois ,Majusc, 86,0,0,                    /* V*/
    87,Une_fois ,Majusc, 87,0,0,                    /* W*/
    88,Une_fois ,Majusc, 88,0,0,                    /* X*/
    89,Une_fois ,Majusc, 89,0,0,                    /* Y*/
   217,Une_fois ,Maj+Op, 89,0,0,                    /* Ÿ*/
    90,Une_fois ,Majusc, 90,0,0,                    /* Z*/
   174,Une_fois ,Maj+Op, 65,0,0,                    /* Æ*/
   206,Une_fois ,Maj+Op, 79,0,0,                    /* Œ*/
    38,Une_fois ,Touche, 38,0,0,                    /* &*/
    97,Une_fois ,Touche, 97,0,0,                    /* a*/
   135,Deux_fois ,Maj+Op, 49,Touche, 97,            /* á*/
   136,Une_fois ,Touche,136,0,0,                    /* à*/
   136,Deux_fois ,Touche, 96,Touche, 97,            /* à*/
   137,Deux_fois ,Touche, 94,Touche, 97,            /* â*/
   138,Deux_fois ,Majusc,172,Touche, 97,            /* ä*/
   139,Deux_fois ,Option,110,Touche, 97,            /* ã*/
   140,Impossible,Touche,140,0,0,                   /* å*/
    64,Une_fois ,Option, 96,0,0,                    /* @*/
   187,Une_fois ,Maj+Op, 85,0,0,                    /* ª*/
    98,Une_fois ,Touche, 98,0,0,                    /* b*/
    99,Une_fois ,Touche, 99,0,0,                    /* c*/
   141,Une_fois ,Touche,141,0,0,                    /* ç*/
   100,Une_fois ,Touche,100,0,0,                    /* d*/
   101,Une_fois ,Touche,101,0,0,                    /* e*/
   142,Une_fois ,Touche,142,0,0,                    /* é*/
   142,Deux_fois ,Maj+Op, 49,Touche,101,            /* é*/
   143,Une_fois ,Touche,143,0,0,                    /* è*/
   143,Deux_fois ,Touche, 96,Touche,101,            /* è*/
   144,Deux_fois ,Touche, 94,Touche,101,            /* ê*/
   145,Deux_fois ,Majusc,172,Touche,101,            /* ë*/
   102,Une_fois ,Touche,102,0,0,                    /* f*/
   103,Une_fois ,Touche,103,0,0,                    /* g*/
   104,Une_fois ,Touche,104,0,0,                    /* h*/
   105,Une_fois ,Touche,105,0,0,                    /* i*/
   146,Deux_fois ,Maj+Op, 49,Touche,105,            /* í*/
   147,Deux_fois ,Touche, 96,Touche,105,            /* ì*/
   148,Deux_fois ,Touche, 94,Touche,105,            /* î*/
   149,Deux_fois ,Majusc,172,Touche,105,            /* ï*/
   245,Une_fois ,Maj+Op, 78,0,0,                    /* ı*/
   106,Une_fois ,Touche,106,0,0,                    /* j*/
   107,Une_fois ,Touche,107,0,0,                    /* k*/
   108,Une_fois ,Touche,108,0,0,                    /* l*/
   109,Une_fois ,Touche,109,0,0,                    /* m*/
   110,Une_fois ,Touche,110,0,0,                    /* n*/
   150,Deux_fois ,Option,110,Touche,105,            /* ñ*/
   111,Une_fois ,Touche,111,0,0,                    /* o*/
   151,Deux_fois ,Maj+Op, 49,Touche,111,            /* ó*/
   152,Deux_fois ,Touche, 96,Touche,111,            /* ò*/
   153,Deux_fois ,Touche, 94,Touche,111,            /* ô*/
   154,Deux_fois ,Majusc,172,Touche,111,            /* ö*/
   155,Deux_fois ,Option,110,Touche,111,            /* õ*/
   191,Une_fois ,Option,136,0,0,                    /* ø*/
   188,Impossible,Touche,188,0,0,                   /* º*/
   112,Une_fois ,Touche,112,0,0,                    /* p*/
   113,Une_fois ,Touche,113,0,0,                    /* q*/
   114,Une_fois ,Touche,114,0,0,                    /* r*/
   115,Une_fois ,Touche,115,0,0,                    /* s*/
   116,Une_fois ,Touche,116,0,0,                    /* t*/
   117,Une_fois ,Touche,117,0,0,                    /* u*/
   156,Deux_fois ,Maj+Op, 49,Touche,117,            /* ú*/
   157,Une_fois ,Touche,157,0,0,                    /* ù*/
   157,Deux_fois ,Touche, 96,Touche,117,            /* ù*/
   158,Deux_fois ,Touche, 94,Touche,117,            /* û*/
   159,Deux_fois ,Majusc,172,Touche,117,            /* ü*/
```

```
118,Une_fois ,Touche,118,0,0,                    /* v */
119,Une_fois ,Touche,119,0,0,                    /* w */
120,Une_fois ,Touche,102,0,0,                    /* x */
121,Une_fois ,Touche,121,0,0,                    /* y */
216,Deux_fois ,Majusc,172,Touche,121,            /* ÿ */
122,Une_fois ,Touche,122,0,0,                    /* z */
190,Une_fois ,Option, 97,0,0,                    /* æ */
207,Une_fois ,Option,111,0,0,                    /* œ */
222,Une_fois ,Option,103,0,0,                    /* fi */
223,Une_fois ,Maj+Op, 71,0,0,                    /* fl */
 63,Une_fois ,Touche, 63,0,0,                    /* ? */
192,Une_fois ,Maj+Op, 63,0,0,                    /* ¿ */
 33,Une_fois ,Touche, 33,0,0,                    /* ! */
193,Une_fois ,Option, 33,0,0,                    /* ¡ */
 44,Une_fois ,Touche, 44,0,0,                    /* , */
 46,Une_fois ,Touche, 46,0,0,                    /* . */
201,Une_fois ,Option, 46,0,0,                    /* ... */
225,Une_fois ,Maj+Op, 70,0,0,                    /* . */
 58,Une_fois ,Touche, 58,0,0,                    /* : */
 59,Une_fois ,Touche, 59,0,0,                    /* ; */
202,Une_fois ,Option, 32,0,0,                    /*  */
 36,Une_fois ,Touche, 36,0,0,                    /* $ */
163,Une_fois ,Majusc,163,0,0,                    /* £ */
180,Impossible,Touche,180,0,0,                   /* ¥ */
 39,Une_fois ,Touche, 39,0,0,                    /* ' */
 34,Une_fois ,Touche, 34,0,0,                    /* " */
227,Impossible,Touche,227,0,0,                   /* „ */
210,Une_fois ,Option, 34,0,0,                    /* " */
211,Une_fois ,Maj+Op, 34,0,0,                    /* " */
212,Une_fois ,Option, 39,0,0,                    /* ' */
213,Une_fois ,Maj+Op, 39,0,0,                    /* ' */
226,Une_fois ,Option, 36,0,0,                    /* , */
199,Une_fois ,Option,143,0,0,                    /* « */
200,Une_fois ,Maj+Op, 55,0,0,                    /* » */
 95,Une_fois ,Majusc, 95,0,0,                    /* _ */
208,Une_fois ,Maj+Op, 45,0,0,                    /* – */
209,Une_fois ,Option, 45,0,0,                    /* — */
 35,Une_fois ,Maj+Op,163,0,0,                    /* # */
 37,Une_fois ,Touche, 37,0,0,                    /* % */
228,Une_fois ,Maj+Op, 37,0,0,                    /* ‰ */
 40,Une_fois ,Touche, 40,0,0,                    /* ( */
 41,Une_fois ,Touche, 41,0,0,                    /* ) */
 91,Une_fois ,Maj+Op, 40,0,0,                    /* [ */
 93,Une_fois ,Maj+Op, 41,0,0,                    /* ] */
123,Une_fois ,Option, 40,0,0,                    /* { */
125,Une_fois ,Option, 41,0,0,                    /* } */
 42,Une_fois ,Touche, 42,0,0,                    /* * */
 43,Une_fois ,Touche, 43,0,0,                    /* + */
177,Une_fois ,Maj+Op, 61,0,0,                    /* ± */
 45,Une_fois ,Touche, 45,0,0,                    /* - */
 47,Une_fois ,Touche, 47,0,0,                    /* / */
218,Impossible,Touche,218,0,0,                   /* / */
 92,Une_fois ,Maj+Op, 47,0,0,                    /* \ */
 60,Une_fois ,Touche, 60,0,0,                    /* < */
 62,Une_fois ,Majusc, 62,0,0,                    /* > */
 61,Une_fois ,Touche, 61,0,0,                    /* = */
173,Une_fois ,Option, 61,0,0,                    /* ≠ */
197,Une_fois ,Option,120,0,0,                    /* ≈ */
176,Une_fois ,Option, 44,0,0,                    /* ∞ */
178,Une_fois ,Option, 60,0,0,                    /* ≤ */
179,Une_fois ,Maj+Op, 62,0,0,                    /* ≥ */
 94,Repetition,Touche, 94,0,0,                   /* ^ */
 94,Une_fois ,Maj+Op, 54,0,0,                    /* ^ */
214,Une_fois ,Option, 58,0,0,                    /* ÷ */
215,Une_fois ,Option,118,0,0,                    /* ◊ */
219,Une_fois ,Maj+Op, 82,0,0,                    /* ◊ */
220,Une_fois ,Option,119,0,0,                    /* ‹ */
221,Une_fois ,Maj+Op, 87,0,0,                    /* › */
198,Une_fois ,Maj+Op, 68,0,0,                    /* Δ */
186,Une_fois ,Maj+Op, 66,0,0,                    /* ∫ */
124,Une_fois ,Maj+Op, 76,0,0,                    /* | */
160,Une_fois ,Option,116,0,0,                    /* † */
194,Une_fois ,Option,108,0,0,                    /* ¬ */
195,Une_fois ,Maj+Op, 86,0,0,                    /* √ */
161,Une_fois ,Majusc,161,0,0,                    /* ° */
162,Une_fois ,Maj+Op, 67,0,0,                    /* ¢ */
164,Une_fois ,Touche,164,0,0,                    /* § */
165,Une_fois ,Maj+Op, 46,0,0,                    /* • */
166,Une_fois ,Option,164,0,0,                    /* ¶ */
168,Une_fois ,Option,114,0,0,                    /* ® */
169,Une_fois ,Option, 99,0,0,                    /* © */
170,Une_fois ,Maj+Op, 84,0,0,                    /* ™ */
167,Une_fois ,Option, 98,0,0,                    /* ß */
181,Une_fois ,Option,109,0,0,                    /* µ */
182,Une_fois ,Option,100,0,0,                    /* ∂ */
183,Une_fois ,Maj+Op, 83,0,0,                    /* ∑ */
184,Une_fois ,Maj+Op, 80,0,0,                    /* ∏ */
185,Une_fois ,Option,112,0,0,                    /* π */
189,Une_fois ,Maj+Op, 81,0,0,                    /* Ω */
196,Une_fois ,Option,102,0,0,                    /* ƒ */
224,Une_fois ,Option,113,0,0,                    /* ‡ */
240,Une_fois ,Option, 38,0,0,                    /* ∂ */
171,Repetition,Maj+Op, 49,0,0,                   /* ´ */
172,Repetition,Majusc,172,0,0,                   /* ´ */
126,Repetition,Option,110,0,0,                   /* ~ */
 96,Repetition,Touche, 96,0,0,                   /* ` */
246,Impossible,Touche,246,0,0,                   /* ´ */
247,Impossible,Touche,247,0,0,                   /* ¯ */
248,Impossible,Touche,248,0,0,                   /* ¯ */
249,Impossible,Touche,249,0,0,                   /* ¯ */
250,Impossible,Touche,250,0,0,                   /* ˙ */
251,Impossible,Touche,251,0,0,                   /* ˚ */
252,Impossible,Touche,252,0,0,                   /* ¸ */
253,Impossible,Touche,253,0,0,                   /* ˝ */
254,Impossible,Touche,254,0,0,                   /* ˙ */
255,Impossible,Touche,255,0,0                    /* ˇ */
};

/*********************************************/
int     Ouvert      = 0;
int     g[4]        = { 0xAA55, 0xAA55, 0xAA55, 0xAA55 };
int     t[9]        = { 9, 10, 12, 14, 18, 24, 36, 48, 54 };
int     Speciale    = 0;
int     nLignes;
int     nPolice;
int     aPolice;
int     tPolice;
int     menuID;

Byte    b[8]        = { 128, 64, 32, 16, 8, 4, 2, 1 };
Byte    s[10]       = { 27, '\t', '\010', '\033', '\r', '\003',
        '\034', '\035', '\036', '\037' };
Byte    lc[253] = {
        '0','1','2','3','4','5','6','7','8','9',13,13,
        '',32,'',13,'',202,'',13,'A','Á','Ä','Å',
        'Á','Ä','Ä','B','C','C','D','E','É','E','È',
        'È','F','G','H','I','Í','Î','Ï','Ì','J','K',
        'L','M','N','Ñ','O','Ó','Ô','Ö','Ò','Ò','Ò','Ø',
        'P','Q','R','S','T','U','Ú','Ù','Ü','Ù','V',
        'W','X','Y','Ý','Z',13,13,'Æ','Œ','&',13,13,
        'a','á','à','â','ä','ã','å','@','*','b','c',
        'ç','d','e','é','è','ê','ë','f','g','h','i',
        'í','ì','î','ï','j','k','l','m','n','ñ',
        'o','ó','ô','ö','ò','õ','ø','º','p','q','r',
        's','š','u','ú','ù','û','ü','v','w','x','y',
        'ý','z',13,13,'æ','œ','fi','fl',13,13,'?','¿',
        '!',';',44,'.','...',':','/',';',';',13,13,'$','£',
        '¥',13,13,39,34,227,210,211,212,213,226,199,
        200,13,13,'_','–','—','#','%','‰','(',')','[',
        ']','{','}',13,13,42,'+','±','-',47,218,92,
        '<','>','=','≠','≈','∞','≤','≥','^','+','0',
        'ø','<','>','Δ','∫','|','†','~','√',13,13,'º',
        'ª','§','*','¶','®','©','™',13,13,'ß','µ','∂',
        'Σ','∏','π','Ω','ƒ','‡','∂',13,13,'´','`','~',
        '´','˙','˝','˚','¸','˛','ˇ','˘' };
char    h[16] = {'0','1','2','3','4','5','6','7','8','9','A',
        'B','C','D','E','F' };

DCtlPtr         dce;
WindowPtr       pFenetre = 0L;
MenuHandle      menu = 0L;
ControlHandle   Ctrl[11];
TEHandle        hTexte;

pascal void Defile();

/*********************************************/
main(bParam, DeviceCE, PointEntree)
cntrlParam      *bParam;
DCtlPtr         DeviceCE;
int             PointEntree;
{
if (DeviceCE->dCtlStorage == 0)
    {
    if (PointEntree == 0)
        {
        SysBeep(1);
        CloseDriver(DeviceCE->dCtlRefNum);
    }
    return(0);
}

dce = DeviceCE;
dce->dCtlFlags &= ~dCtlEnable;

switch (PointEntree)
    {
    case 0:
        {
        Ouvrir();
    }
    break;
    case 2:
        {
        rControl(bParam->csCode, bParam->csParam);
    }
    break;
    case 4:
        {
        Fermer();
    }
    break;
    default: ;
}

dce->dCtlFlags |= dCtlEnable;
return(0);
}

/*********************************************/
Ouvrir()
{
dce->dCtlFlags |= dNeedLock|dNeedGoodBye;
if (pFenetre) SelectWindow(pFenetre);
if (Ouvert)
    {
    dce->dCtlMenu = menuID;
    return;
}

Ouvert = 1;

InitMenu();
OuvreFenetre();
}

/*********************************************/
rControl(code, parametres)
int code;
int *parametres;
{
    switch (code)
        {
        case accEvent:
            Evenement(*((EventRecord **)parametres));
            break;
        case accCursor:
            Curseur();
            break;
        case accMenu:
            TrMenu(parametres[1]);
            break;
        case accCut:
            Couper();
            break;
        case accCopy:
            Copier();
            break;
        case accPaste:
            Coller();
            break;
        case accClear:
            Effacer();
            break;
        case goodBye:
            AuRevoir();
            break;
        default: ;
    }
}

/*********************************************/
Fermer()
{
TEDispose(hTexte);
DeleteMenu(dce->dCtlMenu);
dce->dCtlMenu = 0;
DisposeMenu(menu);
DrawMenuBar();
DisposeWindow(pFenetre);
pFenetre = 0L;
dce->dCtlWindow = 0;
}

/*********************************************/
Evenement(event)
EventRecord *event;
{
switch (event->what)
    {
    case updateEvt:
        MiseJour();
        break;
    case mouseDown:
        Clic(event->where, event->modifiers & shiftKey);
        break;
    case activateEvt:
        Active((Boolean)(event->modifiers & activeFlag));
        break;
    case keyDown:
    case autoKey:
        {
        if (event->modifiers & cmdKey )
            Commande((char)(event->message &
            charCodeMask));
        else
            {
            Frappe((char)(event->message &
                charCodeMask));
            AfficheCode(Speciale);
            break;
        }
    }
}
}

/*********************************************/
CalcResource(n)
{
return( 0xC000 + ((~(dce->dCtlRefNum))<<5) + n );
}

/*********************************************/
MiseJour()
{
Rect        r;
register    int i;
GrafPtr     savePort;
```

```
Str255        c;

GetPort( &savePort );
SetPort( pFenetre );

BeginUpdate(pFenetre);
TextFont(geneva);
TextSize(10);

SetRect(&r, 8, 8, 234, 241);
FrameRect(&r);

for (i = 8 ; i < 153 ; i += 16)
    {
    SetRect(&r, 812, i, 385, i + 17);
    FrameRoundRect(&r, 4, 4);
}

PenPat(&g[0]);
SetRect(&r, 256, 8, 305, 169);
FrameRect(&r);
SetRect(&r, 256, 176, 385, 241);
FrameRect(&r);
PenNormal();

MoveTo( 267, 192);
DrawString("\pDécimal :");
MoveTo( 267, 205);
DrawString("\pHexadécimal :");
MoveTo( 267, 218);
DrawString("\pOctal :");
MoveTo( 267, 231);
DrawString("\pBinaire :");

MoveTo( 331, 20);
if (Speciale == 1) TextMode(srcBic);
DrawString("\pEscape");
TextMode(srcOr);
MoveTo( 321, 36);
if (Speciale == 2) TextMode(srcBic);
DrawString("\pTabulation");
TextMode(srcOr);
MoveTo( 318, 52);
if (Speciale == 3) TextMode(srcBic);
DrawString("\p'Back space'");
TextMode(srcOr);
MoveTo( 322, 68);
if (Speciale == 4) TextMode(srcBic);
DrawString("\pAnnulation");
TextMode(srcOr);
MoveTo( 332, 84);
if (Speciale == 5) TextMode(srcBic);
DrawString("\pRetour");
TextMode(srcOr);
MoveTo( 332, 100);
if (Speciale == 6) TextMode(srcBic);
DrawString("\pEntrée");
TextMode(srcOr);
MoveTo( 332, 116);
if (Speciale == 7) TextMode(srcBic);
DrawString("\pGauche");
TextMode(srcOr);
MoveTo( 333, 132);
if (Speciale == 8) TextMode(srcBic);
DrawString("\pDroite");
TextMode(srcOr);
MoveTo( 337, 148);
if (Speciale == 9) TextMode(srcBic);
DrawString("\pHaut");
TextMode(srcOr);
MoveTo( 340, 164);
if (Speciale == 10) TextMode(srcBic);
DrawString("\pBas");
TextMode(srcOr);

AfficheCode(Speciale);

DrawControls(pFenetre);

SetRect(&r, 9, 9, 233, 240);
EraseRect(&r);
TEUpdate(&pFenetre->portRect, hTexte);

EndUpdate( pFenetre );

SetPort( savePort );
}

/*********************************************/

Curseur()
{
Point          pt;
Rect           r;
CursHandle     h;
GrafPtr        savePort;

GetPort(&savePort);
SetPort(pFenetre);

SetRect(&r, 9, 9, 233, 240);
TEIdle(hTexte);
GetMouse(&pt);
if (PtInRect(pt, &r))
    {
    h = GetCursor(iBeamCursor);
    SetCursor(*h);
}
else   InitCursor();
SetPort(savePort);
}

/*********************************************/
Couper()
{
SpecialeOff();
TECut(hTexte);
Selection();
}

/*********************************************/
Copier()
{
SpecialeOff();
TECopy(hTexte);
Selection();
}

/*********************************************/
Coller()
{
SpecialeOff();
TEPaste(hTexte);
Selection();
}

/*********************************************/
Effacer()
{
SpecialeOff();
TEDelete(hTexte);
Selection();
}

/*********************************************/
Clic(p,s)
Point      p;
int        s;
{
GrafPtr            savePort;
int                cntlCode;
int                b, i;
ControlHandle      Controle;
Rect               r;

GetPort(&savePort);
SetPort(pFenetre);
GlobalToLocal(&p);

if (cntlCode = FindControl(p, pFenetre, &Controle))
    {
    if (GetCRefCon(Controle) == nScroll)
        {
        if (cntlCode == inThumb)
            {
            TrackControl(Controle, p, 0L);
            Ajuste();
        }
        else
            {
            TrackControl(Controle, p, &Defile);
        }

    }
    else
        {
        if (TrackControl(Controle, p, 0L))
        b = GetCRefCon(Controle);
        if (b == pComp)
            {
            SetRect(&r, 9, 9, 233, 240);
            EraseRect(&r);
            TEDispose(hTexte);
            PoliceComplete();
        }
        else
            {
            for (i = 0 ; i < 9 ; i++)
                {
                SetCtlValue(Ctrl[i], Non);
            }
            SetCtlValue(Ctrl[b], Oui);
            tPolice = t[b];
            ChangePolice();
        }
    }
}
else
    {
    if (PtInRect(p, &(**hTexte).viewRect))
        {
        SpecialeOff();
        TEClick(p, s != 0, hTexte);
        AfficheCode(Speciale);
        Selection();
    }
    else SelSpeciale(p);
}
SetPort(savePort);
}

/*********************************************/
Active(a)
Boolean a;
{
if (a)
    {
    TEActivate(hTexte);
    TEFromScrap();
    InsertMenu(menu, 0);
    DrawMenuBar();
    if(Speciale)      DesSpeciale();
}
else
    {
    if(Speciale)      DesSpeciale();
    TEDeactivate(hTexte);
    ZeroScrap();
    TEToScrap();
    DeleteMenu(dce->dCtlMenu);
    DrawMenuBar();
}
}

/*********************************************/
AuRevoir(a)
Boolean a;
{
}

/*********************************************/
Frappe(c)
char c;
{
SpecialeOff();
TEKey(c, hTexte);
Selection();
}

/*********************************************/
OuvreFenetre()
{
GrafPtr      savePort;
register     int    i;
Rect         r;
static       Rect   Fen = { 50, 8, 322, 400 },
                    scr = { 8, 233, 241, 249 },
                    textV  = { 9, 9, 240, 232 },
                    text   = { 12, 12, 237, 229 },
                    rb  = { 16, 262, 32, 298 };

GetPort(&savePort);

pFenetre = NewWindow( 0L, &Fen, "\pPom's - Stricto
    sensu", Oui, noGrowDocProc, -1L, Oui, 0L );

((WindowPeek)pFenetre)->windowKind =
    dce->dCtlRefNum;
dce->dCtlWindow = pFenetre;

SetPort(pFenetre);

Ctrl[9] = NewControl(pFenetre, &scr, "\p", Oui, 0, 0, 0,
    scrollBarProc, nScroll);
SetRect(&r, 256, 248, 385, 265);
Ctrl[10] = NewControl(pFenetre, &r, "\pPolice
Complète", Oui, 0, 0, 1, pushButProc, pComp);

Ctrl[0] = NewControl(pFenetre, &rb, "\p9", Oui, 0, 0, 1,
    radioButProc, t9);
rb.top += 16;
rb.bottom += 16;
Ctrl[1] = NewControl(pFenetre, &rb, "\p10", Oui, 0, 0, 1,
    radioButProc, t10);
rb.top += 16;
rb.bottom += 16;
Ctrl[2] = NewControl(pFenetre, &rb, "\p12", Oui, 0, 0, 1,
    radioButProc, t12);
rb.top += 16;
rb.bottom += 16;
Ctrl[3] = NewControl(pFenetre, &rb, "\p14", Oui, 0, 0, 1,
    radioButProc, t14);
rb.top += 16;
rb.bottom += 16;
Ctrl[4] = NewControl(pFenetre, &rb, "\p18", Oui, 0, 0, 1,
    radioButProc, t18);
rb.top += 16;
rb.bottom += 16;
Ctrl[5] = NewControl(pFenetre, &rb, "\p24", Oui, 0, 0, 1,
    radioButProc, t24);
rb.top += 16;
rb.bottom += 16;
Ctrl[6] = NewControl(pFenetre, &rb, "\p36", Oui, 0, 0, 1,
    radioButProc, t36);
rb.top += 16;
rb.bottom += 16;
Ctrl[7] = NewControl(pFenetre, &rb, "\p48", Oui, 0, 0, 1,
    radioButProc, t48);
```

```
rb.top += 16;
rb.bottom += 16;
Ctrl[8] = NewControl(pFenetre, &rb, "\p54", Oui, 0, 0, 1,
    radioButProc, t54);

Tailles(nPolice);

PoliceComplete();
SetPort(savePort);
}

/*********************************************/
InitMenu()
{
dce->dCtlMenu = menuID = CalcResource(0);
menu = NewMenu(menuID, "\pPolice");
AddResMenu(menu, 'FONT');
Defaut();
}

/*********************************************/
TrMenu(n)
int          n;
{
Str255       cArticle;
CheckItem(menu, aPolice, Non);
aPolice = n;
CheckItem(menu, aPolice, Oui);
GetItem(menu, aPolice, &cArticle);
GetFNum(&cArticle, &nPolice);
Tailles(nPolice);
ChangePolice();
HiliteMenu(0);
}

/*********************************************/
Defaut()
{
register  int    Compteur;
register  int    i;
Str255           cArticle;
Compteur = CountMItems(menu);
for (i = 1 ; i < (Compteur + 1) ; i++)
    {
    aPolice = 0;
    GetItem(menu, i, &cArticle);
    if (IUCompString("\pTimes", &cArticle) == 0)
        {
        aPolice = i;
        break;
    }
}
if (aPolice)
    {
    CheckItem(menu, aPolice, Oui);
    GetItem(menu, aPolice, &cArticle);
    GetFNum(&cArticle, &nPolice);
}
else
    {
    for (i = 1 ; i < (Compteur + 1) ; i++)
        {
        GetItem(menu, i, &cArticle);
        if (IUCompString("\pChicago", &cArticle) == 0)
            {
            aPolice = i;
            break;
        }
    }
CheckItem(menu, aPolice, Oui);
GetItem(menu, aPolice, &cArticle);
GetFNum(&cArticle, &nPolice);
}
}

/*********************************************/
Tailles(nP)
register  int    nP;
{
register  int    i;
for (i = 0 ; i < 9 ; i++)
    {
    SetCtlValue(Ctrl[i], Non);
    if (RealFont(nP, t[i]))
        {
        HiliteControl(Ctrl[i], 0);
    }
    else
        {
        HiliteControl(Ctrl[i], 255);
    }
}
tPolice = 0;
for (i = 5 ; i > -1 ; i--)
    {
    if (RealFont(nP, t[i]))
        {
        SetCtlValue(Ctrl[i], Oui);
        tPolice = t[i];
        break;
    }
}
if (tPolice == 0)
for (i = 6 ; i < 9 ; i++)
    {
    if (RealFont(nP, t[i]))
        {
        SetCtlValue(Ctrl[i], Oui);
        tPolice = t[i];
        break;
    }
}
}

/*********************************************/
ChangePolice()
{
Rect         r;
FontInfo     f;
GrafPtr      savePort;

GetPort(&savePort);
SetPort(pFenetre);
SpecialeOff();
TextFont(nPolice);
TextSize(tPolice);
GetFontInfo(&f);
(**hTexte).txFont = nPolice;
(**hTexte).txSize = tPolice;
(**hTexte).fontAscent = f.ascent;
(**hTexte).lineHeight = f.ascent + f.descent;
Vue();
Selection();

SetRect(&r, 9, 9, 233, 240);
InvalRect(&r);
SetPort(savePort);
}

/*********************************************/
Selection()
{
register  int    LigneHaut, LigneBas, Ligne;

SetVScrollEdit();
Ajuste();

LigneHaut = GetCtlValue(Ctrl[9]);
LigneBas = LigneHaut + nLignes;

if ((**hTexte).selStart <
    (**hTexte).lineStarts[LigneHaut] ||
    (**hTexte).selStart >=
    (**hTexte).lineStarts[LigneBas]) {
    for (Ligne = 0; (**hTexte).selStart >=
        (**hTexte).lineStarts[Ligne]; Ligne++);
    SetCtlValue(Ctrl[9], Ligne - nLignes / 2);
    Ajuste();
    }
}

/*********************************************/
SetVScrollEdit()
{
register int   n;

n = (**hTexte).nLines-nLignes;

if ((**hTexte).teLength > 0 &&
    (*((**hTexte).hText))[(**hTexte).teLength-1]=='\r')
    n++;

SetCtlMax(Ctrl[9], n > 0 ? n : 0);
}

/*********************************************/
Ajuste()
{
int     oldScroll, newScroll, delta;

oldScroll = (**hTexte).viewRect.top -
    (**hTexte).destRect.top;
newScroll = GetCtlValue(Ctrl[9]) *
    (**hTexte).lineHeight;
delta = oldScroll - newScroll;
if (delta != 0) TEScroll(0, delta, hTexte);
}

/*********************************************/
Vue()
{
nLignes =((**hTexte).viewRect.bottom-
    (**hTexte).viewRect.top)/(**hTexte).lineHeight;
TECalText(hTexte);
}

/*********************************************/
pascal void Defile(Controle, Code)
ControlHandle Controle;
int          Code;
{
int     TailleP;
int     DefileA;

if (Code == 0)
    return ;

TailleP = ((**hTexte).viewRect.bottom-
    (**hTexte).viewRect.top) /
    (**hTexte).lineHeight - 1;

switch (Code) {
    case inUpButton:
        DefileA = -1;
        break;
    case inDownButton:
        DefileA = 1;
        break;
    case inPageUp:
        DefileA = -TailleP;
        break;
    case inPageDown:
        DefileA = TailleP;
        break;
}
SetCtlValue( Controle, GetCtlValue(Controle)+
    DefileA );
Ajuste();
}

/*********************************************/
AfficheCode(n)
Byte   n;
{
GrafPtr savePort;
Rect       r;
Str255     code;
Byte       v, w, x;
int        i, j;

GetPort(&savePort);
SetPort(pFenetre);

TextFont(geneva);
TextSize(10);

if (n)  v = s[n - 1];
else
    {
    if ((**hTexte).selStart == (**hTexte).selEnd)
        {
        if ((**hTexte).selStart == 0)
            v = 0;
        else
            v = *(*((**hTexte).hText) + (**hTexte).selStart -
                1);
    }
    else
        v = *(*((**hTexte).hText) + (**hTexte).selStart);
}
NumToString((long)v, &code);
SetRect(&r, 317, 184, 380, 193);
EraseRect(&r);
MoveTo(317, 192);
DrawString(&code);

code[0] = 2;
w = v / 16;
code[1] = h[w];
code[2] = h[v - (w * 16)];
SetRect(&r, 338, 197, 380, 206);
EraseRect(&r);
MoveTo(338, 205);
DrawString(&code);

code[0] = 3;
x = v / 64;
code[1] = h[x];
w = (v - (64 * x)) / 8;
code[2] = h[w];
code[3] = h[v - (64 * x) - (8 * w)];
SetRect(&r, 302, 210, 380, 219);
EraseRect(&r);
MoveTo(302, 218);
DrawString(&code);

code[0] = 9;
code[5] = ' ';
for (x = 0 ; x < 8 ; x++)
    {
    w = v;
    if (x > 3)  i = 2;
    else            i = 1;
    if (w & b[x])   code[x + i] = '1';
    else            code[x + i] = '0';
}
SetRect(&r, 315, 223, 380, 232);
EraseRect(&r);

MoveTo(315, 231);
DrawString(&code);

SetRect(&r, 8, 246, 248, 264);
FraseRect(&r);

if (v > 31)
    {
```

A l'essai :



ConcertWare

Christian Piard

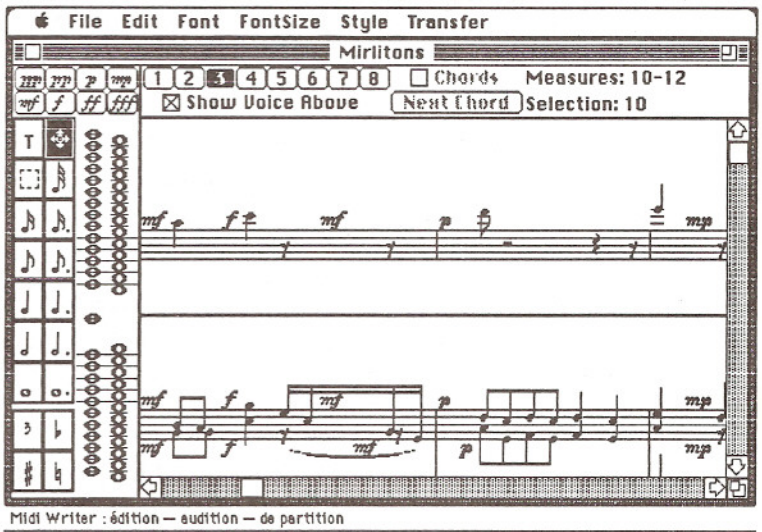
ConcertWare+ est un logiciel musical pour Macintosh composé de trois applications : la première permet de créer et de travailler les partitions, la seconde est conçue pour l'édition du son des instruments, la troisième joue l'œuvre. De l'édition des partitions au pilotage d'un instrument électronique, l'ensemble est étonnant.

Notre version 4.05 de ConcertWare+ a été testée sur un Macintosh II et un Macintosh Plus connecté par l'interface MIDI à un piano numérique Yamaha. Précisons d'emblée qu'un instrument MIDI n'est pas indispensable pour l'utilisation de ConcertWare+ même si c'est un prolongement naturel ; il existe en effet deux versions : ConcertWare+ et ConcertWare+ MIDI.

Présentation

ConcertWare+ est livré avec une documentation demi-format d'une centaine de pages en anglais bien entendu, plus un fascicule de vingt pages pour la version MIDI. Si l'ensemble est bien fait, bien présenté et agrémenté de copies d'écran, il n'en reste pas moins qu'une version française simplifierait les choses. Même si le maniement est assez intuitif, le recours à la documentation s'impose.

On regrette que l'aspect MIDI soit réduit au nécessaire, malgré le fichier texte qui figure sur le disque. Il y a tout ce qu'il faut pour s'en sortir, mais il n'y a rien de trop.



Midi Writer : édition — audition — de partition

```
    MoveTo(8, 260);
    for (i = 0 ; i < 231 ; i++)
        {
        if (v == st[i * 6])    break;
    }
    j = 0;
Boucle:
    switch (st[(i * 6) + 1 + j])
        {
        case Impossible:
            {
            DrawString("\pUtilisez le 'Copier/Coller'");
        }
        break;
        case Une_fois :
            {
            switch (st[(i * 6) + 2 + j])
                {
                case 0:
                    {
                    DrawChar('");
                    DrawChar(st[(i * 6) + 3 + j]);
                    DrawChar("');
                    DrawString("\p seul");
                }
                break;
                case 1:
                    {
                    DrawString("\pMajuscule ");
                    DrawChar('");
                    DrawChar(st[(i * 6) + 3 + j]);
                    DrawChar("');
                }
                break;
                case 2:
                    {
                    DrawString("\pOption ");
                    DrawChar('");
                    DrawChar(st[(i * 6) + 3 + j]);
                    DrawChar("');
                }
                break;
                case 3:
                    {
                    DrawString("\pOption - Majuscule ");
                    DrawChar('");
                    DrawChar(st[(i * 6) + 3 + j]);
                    DrawChar("');
                }
                break;
            }
        }
        break;
        case Deux_fois :
            {
            switch (st[(i * 6) + 2 + j])
                {
                case 0:
                    {
                    DrawChar('");
                    DrawChar(st[(i * 6) + 3 + j]);
                    DrawChar("');
                    DrawString("\p, puis ");
                    switch (st[(i * 6) + 4 + j])
                        {
                        case 0:
                            {
                            DrawChar('");
                            DrawChar(st[(i * 6) + 5 + j]);
                            DrawChar("');
                        }
                        break;
                        case 1:
                            {
                            DrawString("\pMajuscule ");
                            DrawChar('");
                            DrawChar(st[(i * 6) + 5 + j]);
                            DrawChar("');
                        }
                    }
                }
                break;
                case 1:
                    {
                    DrawString("\pMajuscule ");
                    DrawChar('");
                    DrawChar(st[(i * 6) + 3 + j]);
                    DrawChar("');
                    DrawString("\p, puis ");
                    switch (st[(i * 6) + 4 + j])
                        {
                        case 0:
                            {
                            DrawChar('");
                            DrawChar(st[(i * 6) + 5 + j]);
                            DrawChar("');
                        }
                        break;
                        case 1:
                            {
                            DrawString("\pMajuscule ");
                            DrawChar('");
                            DrawChar(st[(i * 6) + 5 + j]);
                            DrawChar("');
                        }
                    }
                }
                break;
                case 2:
                    {
                    DrawString("\pOption ");
                    DrawChar('");
                    DrawChar(st[(i * 6) + 3 + j]);
                    DrawChar("');
                    DrawString("\p, puis ");
                    switch (st[(i * 6) + 4 + j])
                        {
                        case 0:
                            {
                            DrawChar('");
                            DrawChar(st[(i * 6) + 5 + j]);
                            DrawChar("');
```

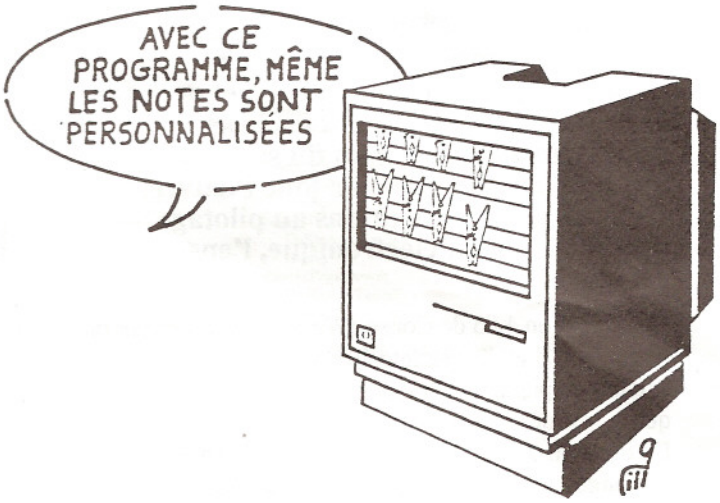
Suite page 59

Disons qu'une connaissance de la norme MIDI n'est pas un handicap à la bonne utilisation de ConcertWare+.

Bon point, les programmes ne sont pas protégés et s'installent sans difficulté sur le disque dur, il suffit d'ajouter au Système la police Sonata livrée sur le même disque. Notons à ce sujet que ConcertWare+ est utilisable sur disquette 3,5" sans que ce soit la galère.

Ce programme fonctionne sur tout Mac avec au minimum 512Ko de RAM, sauf sur le Mac XL (Lisa). En l'absence d'instrument MIDI, la connexion à votre chaîne HiFi est une quasi-nécessité, le haut-parleur intégré au Mac étant plutôt un 'bas-parleur' qui ne brille pas par sa fidélité.

ConcertWare+ se compose de trois programme principaux : **MidiWriter** pour l'écriture des partitions, **MidiPlayer** pour l'exécution des musiques et **Instrument Maker** pour modifier l'orchestre (un menu offre le passage direct d'une application à l'autre sans retour au Finder).



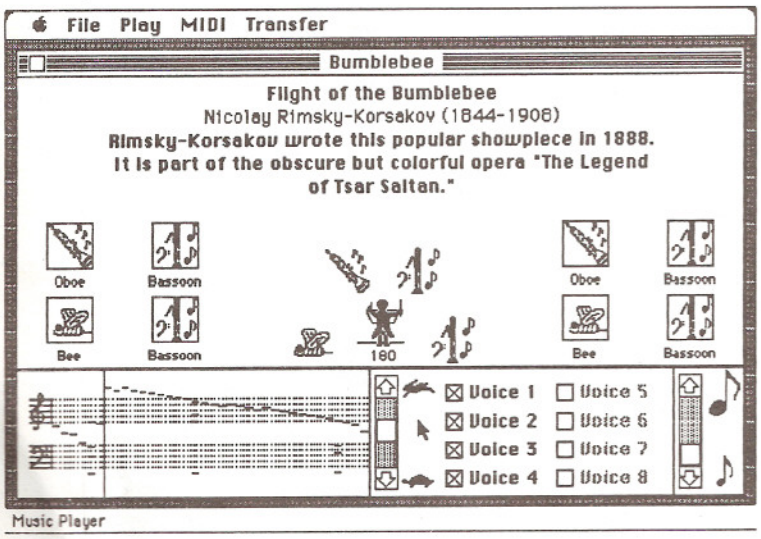
MidiWriter : la partition

L'écriture d'une partition avec MidiWriter tient du jeu d'enfant, le souci des programmeurs ayant clairement été la simplification : au choix, les notes seront saisies à l'aide de la souris, du clavier du Macintosh ou du clavier MIDI connecté. En ce qui nous concerne, le clavier du Yamaha a bien servi.

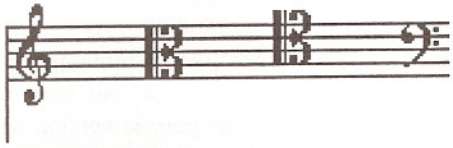
À tout instant, il sera possible d'entendre sur le Mac —ou sur l'instrument MIDI connecté— la partition, un morceau de partition ou même telle et/ou telle voie.

Les possibilités

Première chose, installer les portées : nous avons huit voies disponibles qu'il est possible d'installer sur 1 à 8 portées ; dans un souci de clarté à l'écran, il est possible de demander par un simple clic l'affichage isolé de la voie sur laquelle on travaille.



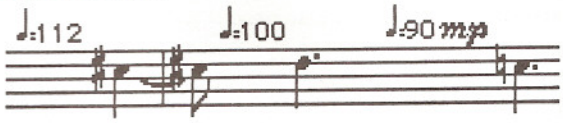
Nous avons les portées, choisissons les clefs :



Les graphismes de la police Sonata sont très satisfaisants à l'écran et l'impression sur une LaserWriter est un régal... Bien que la LaserWriter ne connaisse pas la police Sonata, l'impression se fait par vecteur et non en *bitmap* comme la copie d'écran ci-dessus.

Bien entendu, il est possible de changer de clef en tout endroit de la partition. Choisir maintenant la tonalité relève à nouveau d'un simple clic, dièses et bémols s'inscrivant alors automatiquement. Idem pour la mesure 2-2 6-8 etc.

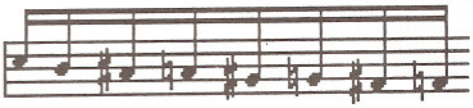
En ce qui concerne le tempo, la souris résoud à nouveau la question, à tout endroit de la partition ; voici par exemple un ralentissement :



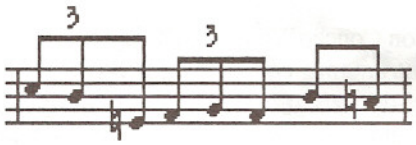
On en vient alors à la saisie des notes, par l'un des trois moyens indiqués. Choisir la durée —croche, noire, etc.— puis l'altération accidentelle éventuelle —dièse, bécare etc.— et enfin la hauteur. La note s'inscrit où il convient avec insertion automatique des barres de mesure en fonction de ce qu'on a indiqué à la clef.

Dans l'une des 19 partitions exemples de ConcertWare+, on trouve des double-croches en quantité, de quoi délier les doigts, il s'agit du Vol du Bourdon de Rimsky Korsakov :

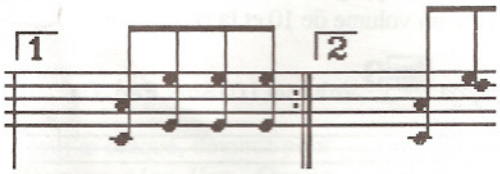
D'accord, présentées ainsi, c'est peu lisible. Qu'à cela ne tienne, un clic relie tout ce petit monde :



Mais oui, les triolets sont aussi prévus, avec une notation standard et s'ils gênent le pianiste débutant, ConcertWare+ nous joue ça en faisant semblant de ne pas souffrir :

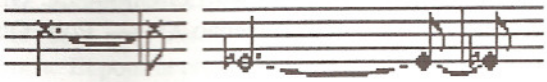


L'essentiel de l'arsenal d'écriture musicale est là, y compris reprise ou da capo. Un exemple tiré de Scott Joplin :



On voit ici que les accords ne posent pas plus de problème à MusicWriter qu'au "programmeur de partition". Plus fort, l'enregistrement peut se faire en temps réel depuis un instrument MIDI, le Mac devient séquenceur et offre un métronome permettant de se caler sur les mesures ; il faut avoir précisé la *résolution* voulue de telle façon que des irrégularités dans le rythme ne conduisent pas à trop de huitièmes de soupir.

Que nous manque-t-il encore, les nuances ? Pas de problème, de *fff* à *ppp* en huit pas et sur votre instrument MIDI en 127 pas !
Côté présentation de la partition, on n'a pas oublié que les notes ne sont pas toujours rondes...



Le MacWrite de la double croche

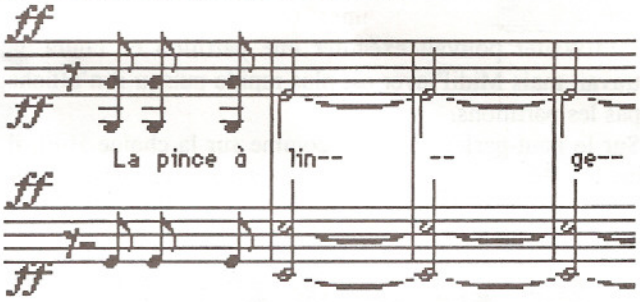
MidiWriter est plus qu'un logiciel à jouer des notes, c'est véritablement un 'traitement de musiques' comme Word est un traitement de textes. Ce programme peut être vu comme un utilitaire de PAO pour travailler les partitions et les éditer, il n'est pas uniquement l'outil du musicien.

Entre le musicien averti qui s'intéresse au fond et l'éditeur qui s'intéresse à la qualité des partitions imprimées, MidiWriter séduira aussi celui qui, sans être musicien, saisit en quelques minutes une partition pour le plaisir des oreilles ou pour lui faire subir tous les outrages.

Notons à ce sujet que J.-S. Bach est un réel plaisir sur micro ; à croire qu'il avait écrit pour les sonorités électroniques du Mac, des synthétiseurs et autres pianos numériques. Le côté "petite horloge bien réglée" va bien à ConcertWare+ : le Mac bien tempéré...

La souplesse du Mac

Si votre œuvre se chante, pas de problème non plus mais là, ConcertWare+ fait plus fort : après avoir choisi la police et le corps du texte, on clique vers la première note et on tape les syllabes attachées à cette note. Puis tabulation pour passer à la suivante, etc. L'espacement des notes variera automatiquement en fonction du texte comme ici entre les trois premières croches :



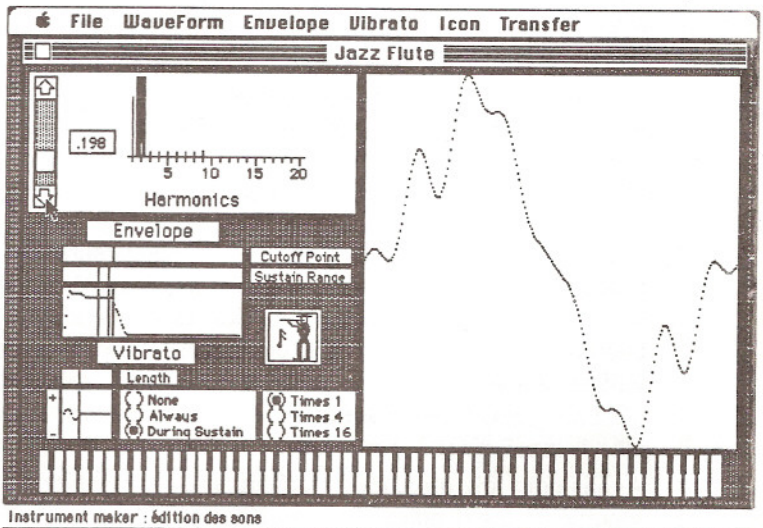
Vous avez dit souplesse ?

Qu'attend-on d'un ordinateur, si ce n'est une simplification : l'ensemble de la partition est saisi horizontalement, les retours à la ligne n'apparaissant que lors de l'impression qui gère la mise en page...

Changer de tonalité, remonter toutes les voies d'un tierce, en descendre une d'une octave ou un demi-ton, copier une voie et la coller plus loin ne sont qu'affaire de clics. Copier toutes les voies et les coller ailleurs mais en les changeant de portées, le clic toujours.

Bien entendu, si vous passez d'une mesure 2-4 à une 3-4 ou autre, ConcertWare+ se charge de rectifier la position des barres de mesure. Si une noire se trouve alors à cheval sur deux mesures, elle sera —en fonction des temps— divisée en une croche pointée en fin de première mesure et une double croche en début de deuxième, avec liaison bien sûr...

Vous avez un Mac II, quelle chance, comme sur la couverture de ce numéro chaque voie peut avoir sa couleur...
Copie d'écran partielle ou complète, décidément on a pensé à tout, rien de manque mais...
Mais on regrettera une certaine lenteur dans le déplacement horizontal sur les portées avec les ascenceurs. Pour aller à la mesure 37, il est préférable d'utiliser le menu *ad hoc* que de scroller note par note ou mesure par mesure. Sur le Mac II, c'est moins gênant.



Instrument maker : édition des sons

Midi Player : l'exécution

MidiWriter pouvait exécuter une partition en cours de travail mais MidiPlayer est plus rapide puisqu'il n'affiche pas les partitions.
Sur le haut-parleur interne, comme sur la chaîne HiFi, il est capable de jouer quatre voies seulement sur les huit. Cette limitation n'existe évidemment pas lorsqu'il pilote un instrument MIDI.

Des icônes apparaissent indiquant les instruments choisis mais, malgré les indications dans la partitions, MidiPlayer laisse la possibilité d'en changer, même en cours d'exécution. Comme un piano numérique n'a pas nécessairement les sonorités que le "programmeur musical" a choisies, les changements d'instruments provoquent parfois des passages inattendus en clavecin ou célesta... La solution est encore là, demander simplement que les changements d'instruments ne soient pas envoyés sur l'interface Midi...

On changera, toujours avec la souris, la vitesse d'exécution, le niveau sonore sur le haut-parleur interne et les voies jouées.

Enfin, la synchronisation avec une boîte de rythmes ou un autre Mac par exemple est assurée : votre Mac sera Maître ou Esclave dans son exécution grâce à la transmission de l'horloge entre les éléments Midi. Pas de cacophonie.

Instrument Maker : la lutherie

Ici se crée l'orchestre.

ConcertWare+ est livré avec 115 instruments, tous éditables avec la souplesse Mac. On peut ainsi créer de nouvelles sonorités qui seront exploitées par MidiPlayer ou MidiWriter.

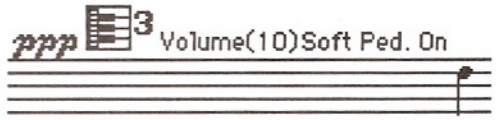
L'édition se fait par dessin direct de l'onde à l'écran et calcul en fonction des harmoniques demandées et de leurs niveaux relatifs. On intervient également sur le vibrato, en durée et en amplitude, sur le "sustain", la variation du niveau sonore en fonction du temps (enveloppe).

Inutile de réinventer l'eau tiède : par le copier/coller, on peut concevoir un instrument en partant de la forme d'onde de la contrebasse travaillée avec les harmoniques du bourdon, avec l'enveloppe de la clarinette, le vibrato de la guitare et une icône basée sur celle du tuba. Le résultat n'est pas garanti concernant cet exemple mais il donne une idée de la puissance de l'ensemble.

Avons-nous dit qu'à tout moment on peut essayer la sonorité sur les 76 notes du clavier à l'écran ?

Liaison MIDI

Sur la version ConcertWare+ MIDI, la gestion de la liaison avec les intruments connectés est aussi simple. Dans l'exemple ci-dessous, on a demandé une 'vélocité' de 8 (sur 127 pas) ce qui nous donne un *ppp* (mais c'est plus précis que l'option *ppp* à l'écran).
On jouera avec l'instrument 3 de notre Yamaha (bien entendu, on peut jongler à loisir sur les numéros de voies MIDI) avec un volume de 10 et la pédale douce...



L'ensemble des 127 "macros" MIDI, autrement dit les séquences hexadécimales à envoyer aux instruments pour les configurer, est éditable de façon simple, les paramètres attachés étant demandés dans une fenêtre de dialogue classique. On peut essayer une macro lors de son édition sans devoir l'insérer dans une partition et la jouer.

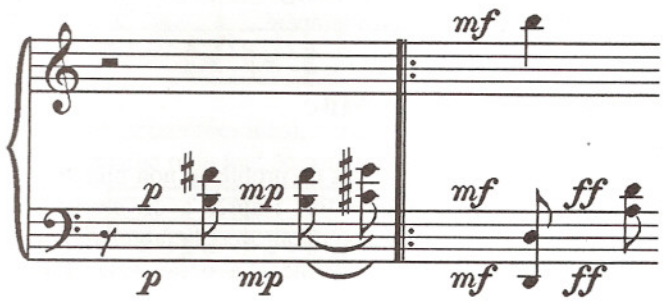
Voici les macros connues de ConcertWare+ lorsque vous le recevez : Play c — All off — Mod. — Breath — Foot — Port. — Data — Volume — Balance — Pan — Expr. — Data Inc. — Data Dec. — Sustain — Sustain Off — Port. On — Port. Off — Sost. On — Sost. Off — Soft Ped. On — Soft Ped. Off — Tremolo On — Tremolo Off — Chorus On — Chorus Off — Celeste On — Celeste Off — Phaser On — Phaser Off — LC On — LC Off — Omni Off — Omni On — Mono On — Poly On — Restore.

Souplesse toujours, on peut décider que les ordres MIDI apparaîtront ou non à l'impression et/ou à l'écran...

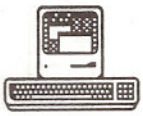
Précisons enfin que deux jeux de paramètres par défaut sont prévus : les très classiques synthétiseurs Casio CZ-101 et Yamaha DX-7.

À 700,00 F la version de base et 1 800,00 F la version Midi, nul doute que le possesseur de Mac un tant soit peu attiré par la musique 'craquera'.

Ayons une pensée émue pour le musicien qui débute, coincé par un rythme qu'il ne sent pas, bien qu'il compte courageusement de 1 à 16 dans chaque mesure : saisir la partition, demander *Play* et entendre enfin ce qu'il fallait comprendre...



Les News



Ariel Sebban

Bon ça y est j'ai craqué, devinez pour quoi ? Pour un Mac IIcx. Quelle merveille ! Y a pas à dire Apple fait fort à chaque nouvelle machine, et là il a fait très fort. Tout y respire la qualité : qualité de son *look*, il est vrai que ça compte et qualité de ses composants internes surtout.

Quand on ouvre un IIcx, on est devant une sorte de "meccano" haut de gamme, où il suffit de dévisser une seule vis pour tout démonter. Et que croyez vous que je fis dès que je le pus ? Je l'ai démonté évidemment. Tout se déboîte sans problème, et surtout tout se remboîte aussi facilement. Et pour la première fois de ma carrière de démonteur invétéré, j'ai pu tout remonter sans qu'il ne reste la pièce dont on ne sait quoi faire, et qui apparemment était en trop ! En fait le contenu se limite à cinq pièces : le disque dur interne, le lecteur de disquette 1,44 Méga, l'alimentation, et la carte-mère, et le haut parleur.

Commençons, par la carte mère : pas le moindre petit fil n'y traîne, c'est propre et limpide. En son centre on y trouve le 68030 cadencé à 15,7 Mhz. Ce processeur a l'avantage par rapport à celui du Mac II de base (68020) de posséder d'emblée la PMMU dont Alain Raynaud vous a parlé dans le précédent numéro. C'est ce co-processeur qui permettra d'utiliser la pleine puissance du système 7, mémoire virtuelle incluse. À lui aussi AU/X ou pour ceux qui ne le sauraient pas Unix version Apple. Multitâche, tout est possible pour ce petit bijou. Et pour épauler le 68030 dans les calculs, vient avec le 68882 le coprocesseur arithmétique. Pour qui a déjà augmenté la mémoire d'un Mac II, le IIcx est ici aussi d'une facilité de mise à jour, étonnante. Plus besoin de tout mettre ventre en l'air pour accéder aux connecteurs de barrettes RAM. À ce propos vous vous souvenez je vous avais prédit une chute spectaculaire de ces composants, c'est fait. On trouve actuellement aux États-Unis des barrettes de un Méga à 135 dollars soit environ 1000 francs. À Paris on commence à 1400 francs.

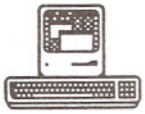
Parlez vous SQL ?

Vous avez certainement entendu parler d'Oracle pour Macintosh. De même 4D, Omnis 5, FoxBase+ Mac, annoncent tous dans leurs produit une interface capable de communiquer avec les bases de données SQL. Apple lui même n'est pas en reste puisqu'il est prévu dans le système 7 une interface CL/1. Alors qu'est ce que tout cela peut bien vouloir dire ?

SQL est l'abréviation de Standard Query Langage. C'est un langage d'interrogation des bases de données Oracle. Ce type de base est devenu un véritable standard puisqu'utilisé par près du quart du parc installé des gros systèmes. Ce langage est quasi indépendant de la machine, et donc universel.

Ainsi un Mac doté de SQL, grâce à une interface adéquate pourra interroger n'importe quelle base de ce type sur n'importe quel système la supportant. Une base de donnée Oracle est organisée non pas en fiches mais en tableaux dont le système gère les liaisons de manière transparente. L'extraction des données se fait en demandant un résultat sans se préoccuper de la démarche à suivre pour l'obtenir. En quelque sorte on demande simplement "je voudrais telle information" sans avoir besoin de définir la procédure pour l'obtention de celle-ci. Ainsi on peut imaginer l'exemple classique d'une liste d'employés occupant un poste dans l'entreprise, et qui perçoivent un salaire. L'organisation de la base se fait donc en une table des employés, une table des postes, et une table des salaires. Chaque fiche est en quelque sorte une ligne de l'accolement de ces tables. L'interrogation des employés touchant un salaire supérieur à x, se fait en demandant 'SELECT "noms" FROM "Fichier" WHERE "salaire" > x'. S'il est vrai que jusqu'ici rien de très différent par rapport aux éditeurs de recherches qu'on trouve de plus en plus sur les micros, la structure réelle du fichier est complètement différente puisqu'en fait chaque table du fichier est totalement indépendante et peut donc appartenir à d'autres fichiers.

La notion elle-même de fichier n'existe plus et l'on parlera plutôt d'ensembles. L'information recherchée étant l'intersection de ces ensembles. Qui a dit que les maths modernes ne servaient à rien ? Ces systèmes d'organisation permettent de gérer d'énormes quantités de données, comme par exemple l'ensemble du fichier des cartes grises sur le territoire français, les recherches se faisant dans des temps étonnants. L'interface SQL, disponible désormais sur les micros dont le Mac permettent d'accéder à cette puissance.



Et cela devrait continuer à chuter d'autant que bientôt les RAMs de 4 Mégas vont commencer à être diffusées en grande quantité. Et le IIcx possédant 8 connecteurs, cela ferait 32 Mégas possibles sur la carte-mère. Jusqu'où irons nous ? Mais ne croyez pas que cela soit inutile, j'ai eu le plaisir de tester des programmes graphique couleur de dernière génération, et dites-vous qu'il ne s'agit que d'un début, et bien ces programmes sont déjà très à l'étroit dans 4 Mégas, si l'on veut en tirer le maximum !

Sur la carte-mère nous trouvons aussi les ROMs soudées et un connecteur de mise à jour de ces ROMs. Apple prétend que cette technique permet d'abaisser les coûts, il faut plutôt y voir le besoin de limiter les risques de marché noir des ROMs du Mac. Et pour compléter le tout, les trois slots NuBus. Pour la grande majorité des utilisateurs, cela suffit largement. Un sera obligatoirement occupé par une carte vidéo, les deux autres vous pourrez en faire ce que vous voudrez. Trois slots suffisent d'autant plus qu'on retrouve aussi les autres connecteurs des deux ports série, deux ports ADB, et le port SCSI. En prime on trouve un connecteur de lecteur de disquette externe, puisque le IIcx, n'accepte pas de deuxième lecteur interne.

À propos du nombre des ports, ce qu'on a reproché longtemps sur le Mac à savoir sa fermeture, sur les machines du monde IBM tous ces connecteurs d'origine sur Mac, sont en option, et donc occupent autant de slots quand ils existent. Que dire encore de cette merveille, que le haut parleur est bien plus grand et procure une qualité de son tout à fait honnête, que le jack haut parleur est stéréo, et qu'enfin l'alimentation de 90 watts est largement dimensionnée pour alimenter les éventuelles cartes additionnelles. L'intérieur du boîtier est complètement recouvert de peinture pour réduire les risques de rayonnements parasites. Sain et propre, telle est véritablement l'impression d'ensemble qu'on peut en retirer lorsqu'on a autopsié un Mac IIcx.

Voilà pour la bête. Et maintenant il faut lui à donner à manger. Je suis donc allé faire un tour à Boston, puisqu'il s'y tenait salon. Cette année, rien de transcendant ne fut présenté à cette MacWorld Expo.

Tout ce qui est à noter vient de partenaires Apple, la pomme se réservant pour Apple Expo qui inaugure cette année le nouveau CNIT à la défense. À ce sujet l'annonce la plus importante sera celle du portable, qui de rêve pour certains devient enfin réalité. Je vous en ai parlé il y a deux numéros, et on j'y reviendrais dès que je l'aurais eu dans les mains. Mais surprise de dernière minute, Apple doit présenter le Mac IIci, qui n'est autre qu'un IIcx cadencé à 25 Mhz, et possédant en série une interface vidéo étendue, laissant libre les trois slots Nubus. Bref un IIcx haut de gamme. Et moi qui pensait qu'avec mon IIcx, j'étais à jour pour au moins 6 mois ! Bon reprenons notre revue de Boston.

Radius : la carte QuickColor Graphics Accelerator

Radius, distribué par P-Ingénierie à Paris, annonce une nouvelle carte graphique pour Macintosh II : la Radius QuickColor Graphics Accelerator.

Grâce à cette carte, QuickDraw 32 bits est accéléré dans un rapport de 1 à 6 pour les opérations graphiques, déplacement de fenêtres, défilement de texte, remplissage de formes, affichages d'images, etc.

L'importante accélération des affichages graphiques a été rendue possible grâce à l'utilisation du microprocesseur RISC VL86C010 et au NuBus Block Transfert (transfert par blocs sur le Nubus) Macintoshien. Rappelons que les processeurs RISC sont dotés de peu d'instructions mais que, s'ils ne savent pas faire grand chose, ils le font vite, très vite...

Le processeur RISC de la carte QuickColor échange les données avec la mémoire vidéo à raison de 27 Mo par seconde, soit 7 fois plus rapidement que le Macintosh II.

La vitesse a un prix : 7 990 FHT.

Les périphériques

Pour les allumés de Space Invaders et autres Flight Simulator, j'ai trouvé un **MouseStick**, qui n'est autre qu'un joystick ou levier de commande et manche à balai pour Mac SE, et toute la famille Mac II, qui se branche sur le bus ADB. Il est vendu 129,95 $US par la société canadienne **Advance Gravis**.

Intéressant aussi le **Mac UnMouse** qui est une tablette digitale compacte à connecter sur un port ADB des Mac SE ou Mac II. On permet le contrôle du curseur, l'entrée des données graphiques et la sélection des touches de fonction. L'utilisation se fait en explorant le verre de la tablette, avec son doigt ou avec un stylo conducteur.

Ses dimensions sont de 7,6 cm X 11,43 cm et la résolution est de 1000 x1000. Prix 690 $US par **Micro-Touch Systems Inc.**

Pour tous les gestionnaires de stocks : le **Bar Code/Magnetic Stripe Combination Readers PC-3000.** C'est un lecteur et encodeur de

code-barre, et de carte magnétique pour Mac SE, et toute la famille Mac II et Apple IIGS utilisant le port ADB. Le périphérique incorpore un microprocesseur qui supprime l'utilisation d'un logiciel de pilotage ce qui le rend très rapide à l'utilisation. Vendu 745 $US l'unité complète par **TPS Electronics**.

Dans le domaine du multimédia, ou le Mac explose on a pu voir entre autres :

La carte RasterOps SFX

C'est une carte 32 bits couleurs QuickDraw pour studio vidéo orienté Multimédia. En mixant les signaux vidéo d'une image graphique du Mac et une source vidéo extérieure, l'utilisateur pourra bien définir le signal dominant de 0 à 255 différents niveaux de dominance (le principe du 32-bit ChromaKeying).

Vidéo Composite, S-Vidéo (S-VHS) en entrée/sortie, RS-343 entrelacé RGB, NTSC, ou PAL en sortie. Cette carte permet aussi le contrôle de la transparence des pixels : pour chaque pixel à l'écran on peut définir un niveau de transparence différent, en utilisant les 8 bits du canal Alpha. Il est possible d'ajuster le signal de sortie afin de faire apparaître la barre de menu du Mac sur l'écran TV. En l'absence du signal genlock le SFX génère un autre signal automatiquement pour une meilleure synchronisation.

Deux modèles de carte étaient proposées : la ColorBoard 232 (32 bits pour une résolution d'écran de 640 X 480) ou la ColorBoard 224 (24 bits pour une résolution d'écran de 1024 X 768). Vendu par **RasterOps**.

Pour voir la vie en rose : 32 bits Color QuickDraw

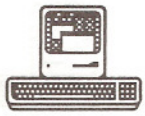
Avec le Mac II est apparue la couleur sur la ligne Macintosh. Jusque là les écrans monochromes de Mac Plus ou SE n'avaient besoin que d'un bit pour définir leur état : allumé ou éteint.

Avec la couleur il a fallu étendre l'information de chaque point d'écran à 2 bits qui donne quatre couleurs, 4 bits pour 16 couleurs et 8 bits pour 256 couleurs. Ce sont ces mêmes 8 bits qui autorisent les 256 niveaux de gris. Pour gérer ce flux de bits il est nécessaire d'avoir une carte vidéo spécialisée qui dispose d'une mémoire à accès rapide. Jusque là QuickDraw n'était capable de gérer que huit bits. Or comme chacun sait le 68020 et 68030 sont des vrai 32 bits et avec 32 bits on peut gérer plus de sept millions de couleurs. Apple vient enfin de mettre à jour les routines de QuickDraw pour exploiter la totalité du bus 32 bits. En fait les cartes vidéo n'utilisent que 24 bits sur les 32 pour la couleur et les huit derniers bits étant destinés à définir un niveau de transparence ou canal Alpha.

Néanmoins Apple n'a pas implémenté les routines nécessaires à la gestion de cette transparence laissant ce soin aux développeurs. De plus le format de fichier procure une importante économie d'espace puisqu'une image peut théoriquement atteindre 2 Mo sur le disque. La technique dite RLL (Run-Length Limited encoding) permet une économie substantielle de mémoire. Les grands éditeurs travaillent sur des nouvelles versions de leurs programmes pour tirer partie de toutes ces nouvelles possibilités. Mais comment faisaient alors les programmes qui avant cette mise à jours annonçaient la gestion de ces 16 millions de couleurs ?

Tout simplement en trichant un peu. Il faut savoir que le système de couleur utilisé par Apple est tout simplement génial en ce sens qu'au lieu d'affecter une couleur à chaque point de l'écran, le système lit d'abord cette valeur dans une table. Et si entre deux rafraîchissements vous modifiez la table, vous obtenez des couleurs par effet de masques qui n'existent pas réellement en tant que telles.

Ce procédé particulièrement intelligent procure au Mac II une rapidité extraordinaire pour ce qui est de la gestion des images.



La carte FrameGrabber 324

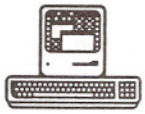
Cette carte vidéo 32 bits couleurs pour la famille Mac II, accepte en entrée des signaux RGB (30 Hz), NTSC, et S-Vidéo et supporte les deux formats 8 bits et 32 bits couleurs. La vitesse de capture de l'image est de l'ordre de 1/30ème de seconde. Le Foyer (focus) est interactif, la digitalisation continue, et on peut sauvegarder ou couper l'image. La carte permet une amélioration importante du contraste d'image, la correction gamma et le réglage de la luminosité. Encore chez RasterOps.

Et toujours chez RasterOps : Une carte qui va faire du bruit : la carte 24 bits couleurs pour Mac SE30. Cette carte est destinée à piloter un écran 13 pouces tel que l'écran haute résolution d'Apple. Si ce genre de carte existe depuis quelque temps déjà la grande nouveauté c'est qu'elle descend au dessous des 1000 dollars rendant désormais les 16 millions de couleurs plus accessibles.

Jusque là ce type de carte coûtait entre trois et cinq mille dollars et en France on en trouve à partir de 30 000 francs HT. Au prix de moins de 7000 francs qui est presque celui de la carte vidéo étendue d'Apple, elle devient un produit de base. Cette carte est de plus entièrement compatible 32 bits Color QuickDraw. Avec ce type de cartes, les couleurs explosent au visage, et on obtient une qualité photo.

MacroMind Director

MacroMind Director permet l'utilisation de plusieurs médias de communication, d'expression ou de divertissements. Par combinaison de



texte, de graphiques, d'animation, de sons et de vidéo, ce système par l'intégration de plusieurs médias est destiné présenter, instruire, visualiser ou simuler des concepts et des idées.

Le package comprend plusieurs modules : MacroMind Accelerator pour augmenter la vitesse de défilement d'images (30 images par seconde) pour approcher la qualité vidéo professionnelle.

MacroMind Director Interactive basé sur une technologie interactive crée des visites guidées sur disque ou disquette. Ainsi pourrait-on développer des simulations ou des prototypes, des kiosques à informations, des disques de formation, des cours d'éducation voire même des logiciels de démonstrations.

MacroMind intègre un puissant logiciel de dessin en couleurs, un contrôleur de palette de couleurs et de l'interface MIDI, une interface utilisateur avec des menus pop-up. MacroMind utilise un langage de scripts se rapprochant beaucoup d'HyperTalk appelé "Lingo" qui permet de créer plusieurs niveaux d'interactivité. Les développeurs pourront établir des liens entre documents et boutons, animer des boutons et les marionnettes, présenter l'ensemble sur des supports vidéo-disques.

Distribué par **P-Ingénierie**.

Pour les Hypercardo-maniaques

Stack Cleaner : c'est un kit d'utilitaires Hypercard comprenant : le Script cleaner qui nettoie les scripts, réduit la taille des piles en intégrant des abréviations standards dans les scripts ; le Split Stack qui permet de sélectionner des cartes et des fonds d'une pile existante pour en créer une nouvelle et le Stack Merge qui facilite la fusion de deux piles pour en faire une. De plus Clone Stack permet de dupliquer la pile originale, en en effaçant le contenu des champs des cartes et des champs de fond. L'utilisateur pourra ainsi détruire facilement les cartes indésirables. Le ResCopy quant à lui copie les ressources d'une application ou d'une pile à une autre pour une copie rapide des menus, des icônes, des sons, et autres ressources. Enfin Menu Maker crée et modifie les menus pour mieux les utiliser dans Hypercard. Des menus peuvent être rajoutés dans la barre de menu ou comme des menus pop-up associés à un champs ou à un bouton. Les menus hiérarchiques sont possibles.

Chez **Trendware Corp.**

HyperX

C'est un outil de système expert pour créer des applications intelligentes, analyser les problèmes ou fournir des conseils de base de l'expertise.
HyperX peut-être combiné avec les multimédia pour résoudre les problèmes de calculs ou de base de données. Interface intelligente de base de données pour mieux analyser les données, les procédures automatiques et les prises de décisions assistées par ordinateur.

Chez **Millienum Software.**

ScriptEdit

Accessoire de bureau intuitif pour créer, modifier, imprimer des scripts d'HyperTalk. Une fonction de comparaison puissante de deux scripts afin de voir leur différence ou leur similitude. Une XCMD pour installer votre pile. **ScriptEdit** augmente la productivité en améliorant l'environnement d'édition des scripts.

Chez **Somak Software Inc.**

HyperTMON

Debuger Interactif du langage de programmation HyperTalk.
HyperTMON apporte les facilités de mise au point avancée pour Hypercard.

Proposé par **Icom Simulations Inc.**

Trendware XCAL

C'est un jeu de XFN et de XCMD qui ajoute des possibilités de feuille de calcul à Hypercard. Les formules, et les macros créées avec **XCAL** sont exécutées automatiquement lors de la saisie de nouvelles données dans les champs. La version internationale est en cours de finalisation.

Chez **Trendware Corporation.**

Les antivirus

AntiToxin est un programme de décontamination et de protection contre les virus Scores, nVir A, nVir B, Hpat, INIT29 et ANTI. AntiToxin est spécialement efficace pour les utilisateurs en réseau. Il protège le Mac des applications infectées du serveur et nettoie le serveur.

Chez **Mainstay.**

VirexGuard

C'est une nouvelle version de Virex qui protège les Mac Plus, SE, et la famille Mac II contre tous virus connus ou non connus : virus Scores, ANTI, MEV#, AIDS, Hpat, INIT29, Peace, et la famille des nVir. **VirexGuard** est un utilitaire très puissant pouvant détecter mêmes les nouveaux virus sur serveur, sur disquette, sur disque dur. VirexGuard répare aussi les fichiers infectés. Une fois le système désinfecté, VirexGuard assure un contrôle permanent du système et vous prévient de toutes infections. Compatible Multifinder, AppleShare, et avec la plupart des logiciels sur réseau.

Chez **HJC Software Inc.**

DoubleHelix

Dans le domaine des bases de données, outre toutes celles que je vous ai récemment annoncées, à savoir 4D nouvelle version, Omnis 5, FoxBase+ Mac V.2, on a pu y voir **DoubleHelix**. Cette base de données

relationnelles sous Mac/OS et pour VMS, que vous connaissez probablement de nom, revient sur le marché français. Quelques caractéristiques intéressantes : nombre illimité de champs, d'enregistrements, de liens entre fichiers, d'icônes, d'index, de nombre de fenêtres, de niveaux de critères de recherche, de nombre total de fichiers dans la base, 128 critères de tri, 32500 caractères par champs, données comprenant textes, nombre, images, date et heure, flags... Ne fait pas le café !
Chez **Odesta Corp**. Sera probablement distribué par G + M Technologies en France.

Voilà succinctement quelques produits qui ne couvrent pas évidement la totalité des nouveautés, mais qui reflètent les grandes orientations du moment.

WinText

Un traitement de textes de plus me direz-vous. Non, tombé par hasard sur la pub de WinText, un traitement de textes d'origine française, j'ai voulu en savoir plus. Allô Messieurs WinSoft, est ce que je pourrais tester pour Pom's votre programme ? Pas de problème Monsieur Sebban nous vous envoyons cela de suite. Et effectivement trois jours plus tard un beau paquet m'était remis pas nos PTT.

Un manuel rouge et blanc et deux disquettes n'attendaient que mes yeux avides. Et je n'ai pas été déçu. Le manuel est clair, aéré et complet. Bon tout cela ne nous dit pas ce que WinText avait en plus. Ce programme est tout d'abord résolument tourné vers 1992. C'est le seul qui gère les césures en quinze langues ! Et quand je dis européen, c'est à l'examen carrément de mondial qu'il faut qualifier WinText. Puisqu'aussi bien le programme gère l'écriture de droite à gauche, comme l'hébreu ou l'arabe. Et l'on peut mixer tous ces types d'écriture dans le même fichier dans un grand élan d'œcuménisme universel.

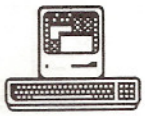
Autre point tout à fait original, est son tableur intégré. On peut définir des cellules contenant des chiffres au milieu de son texte et les relier par des formules comme dans un tableur, à cette liberté près qu'on est pas contraint par un cadre rigide de lignes et colonnes. J'ai répertorié 51 fonctions des plus diverses depuis les fonctions manipulant les dates, jusqu'à la fonction inédite à ma connaissance dans un traitement de textes, qui permet de transformer un nombre en lettres.

Une autre originalité tout à fait intéressante est l'éditeur de formules mathématiques intégré. Ici pas besoin de jongler pour écrire une intégrale ou une racine carrée. C'est du vrai *tel écran tel écrit*. Pour terminer ce tour d'horizon WinText permet de faire du mailing de manière très puissante grâce à ses fonctions conditionnelles.

Au chapitre des regrets, l'absence de multicolonnage, absence d'une fonction de permutation de caractères, qui permet de corriger facilement des fautes de frappes courantes qui consistent en l'inversion de deux lettres dans un mot. Absence aussi de dictionnaire intégré, et faiblesse de l'ouverture dans d'autres formats que ce soit en lecture –ou seul MacWrite est disponible– ou en écriture. Pour autant à 2200 francs on peut lui mettre quatre belle pommes.

Stuffit nouveau

Un shareWare disparaît pour une commercialisation classique : le stuffit nouveau est sorti : c'est **Stuffit deLuxe** s'il vous plaît. Le petit génie qui a écrit le standard des programmes de compactage de données à savoir Stuffit, vient de signer avec **Alladin Systems** qui va s'occuper dorénavant de supporter le produit. Raymond Lau, puisqu'il s'agit de lui, venant de quitter le lycée pour la fac, n'a plus le temps de faire ce travail. Car c'est à 16 ans que ce jeune homme a écrit la première version. Désormais Stuffit s'appellera Stuffit deLuxe, et avec comme amélioration la possibilité de compacter/décompacter en format ARC qui est le standard de compression dans le monde IBM. De plus le programme possède un détecteur



éradiquateur de virus incorporé. Ce qui est sûr c'est que le format de compression proprement dit ne changera pas, un standard échappant souvent à son auteur, les futures versions devraient garder la base Stuffit.

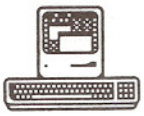
Export

Un produit original a retenu mon attention récemment. Il s'agit d'**Intelligent Developper d'HyperPress**.
Il s'agit d'un système expert. Avec cette particularité d'être 'interfacé' avec HyperCard.

Il y a un ou deux ans on ne jurait plus que par les systèmes experts. Ils devaient révolutionner le monde, et permettre à tout un chacun d'accéder à La Connaissance. La réalité a été toute autre et s'il est vrai que ce type de

Le papier digital ?

Les Ingénieurs de Iomega ont mis au point un prototype de WORM d'un prix tellement faible que d'aucuns n'hésitent pas l'appeler le "papier digital". Ce disque utilise le principe de Bernouilli pour le support. Cette technique autorise l'emploi d'un système optique très bon marché et un laser de très faible puissance pour écrire sur une surface plastifiée de prix insignifiant. Les premières estimations tournent autour de 50 $ pour 1,5 gigabyte (vous avez bien lu). De plus la technique Bernouilli, permet d'atteindre des temps d'accès de l'ordre de ceux de bons disques durs. Si véritablement ce produit sort sur le marché ce sera la révolution de ces prochaines années.



produits trouvent des applications très importantes dans le domaine industriel ou médical, Pour ce qui du *vulgum pecus* que je suis, je n'en ai pas vu la moindre trace d'une utilisation personnelle possible. Interface austère, difficulté d'apprentissage, voir prix exorbitant tout concourait à faire que nous n'en fassions rien.

Et puis en faisant le marché de ce mois ci je suis tombé sur **Intelligent Developper**. C'est un ensemble de développement de systèmes experts qui peut permettre de sortir des applications autonomes, et surtout transformer en pile HyperCard un système créé avec le noyau. La base de connaissance est une base classique à programmer en 'si alors sinon'. Intelligent Developper accepte de multiples formats d'entrée textes, dates, nombres, réponses oui/non etc. Et si l'interface n'est pas fabuleuse, elle fonctionne.

Mais là ou le produit prend toute sa valeur c'est quand une base est terminée, on la transforme en une pile HyperCard. Et là c'est de la magie. Sous les yeux on voit se monter une belle pile avec champs boutons qu'il est possible de personnaliser par la suite. De base de connaissance rébarbative on accède à un produit convivial, qu'on peut mettre entre toutes les mains. Le débouché de ce type de produit devient dès lors évident puisqu'outre les secteurs traditionnels, il devient accessible aux enseignants de lycées, qui pourront y trouver une manière attractive de supports d'exercices dans toutes les disciplines.

Et aussi...

Beaucoup d'autres produits sont arrivés en France récemment : Pour ne citer que quelques uns, je ne citerais que **SUM II** une nouvelle version de SUM testé dans le précédent numéro de Pom's, SUM qui est un kit antivirus du même éditeur Symantec.

J'aurais aimé vous parler plus en profondeur des derniers produits Claris tel que MacDraw II, MacWrite II, ClarisCAD, ainsi que de FileMaker II, mais il semble que les services de presse de BR Publishing, et P-Ingienerie ne daignent pas nous prêter ces programmes pour que nous puissions les tester pour vous. Nous tâcherons donc de nous les procurer directement à la source. Il en est de même pour un distributeur parisien que je ne nommerais pas mais qui représente une quantité suffisamment incompétente pour que nous puissions nous en passer.

Je me suis donc adressé à MacSell chez qui j'ai reçu un accueil plus que chaleureux et efficace pour somme toute les mêmes produits. Il y en a qui comprennent que parler des produits qu'ils importent ne peut que leur faire du bien ! Je pense qu'un jour je ferais un tableau des importateurs/distributeurs qui ne prennent pas les lecteurs de Pom's pour quantité négligeable, nonobstant la qualité de leurs produits qui dans l'ensemble est à de très rares exceptions près de très haut niveau.

DAtabase

Et c'est ainsi que j'ai pu me procurer l'utilitaire du mois à savoir : ce mois-ci un gestionnaire de fichiers en accessoire de bureau : **DAtabase**.

Il est révolu le temps où les accessoires de bureaux n'étaient que des gadgets. Ils sont devenus de véritables applications. L'utilitaire de ce mois est assez exceptionnel. Il permet de disposer d'un véritable gestionnaire de fichier disponible à tout instant. Certes sous multifinder il est toujours possible d'en avoir un ouvert en permanence mais mémoire, quand tu n'est pas là... Ce programme se divise en trois modules : DAtabase Builder est le programme qui fabrique la structure, qui sera utilisée par l'accessoire de bureau proprement dit.

Il sera possible de récupérer des données en format texte par DAtabase Converter.

Les rubriques manipulables par DAtabase sont de type texte, numériques ou même calculées. On peut aussi définir des cases à cocher qui se comportent en rubriques booléennes.

Et pour finir la revue, le programme accepte des rubriques images de type MacPaint ou Pict en couleurs. On peut aussi coller une image en fond de fiche.

Les données peuvent êtres extraites suivant sept critères sur un seul champ. Il est possible de marquer les fiches trouvées pour procéder à une recherche sur un autre champ parmi les fiches marquées. Pour clore cette revue trop rapide de ses étonnantes possibilités, le programme autorise l'export de données sans problèmes.

Avec pour limite 2500 fiches de 50 rubriques, DAtabase, est véritablement un chef d'œuvre comme on aimerait en voir plus souvent. Distribué par **MacSell**, pour un prix d'environ 800 francs, rapport qualité prix hors pair.

J'ai profité de MacSell pour emprunter et tester deux petits produits Hypercard, qui améliorent grandement les fonctions d'édition papier. Il s'agit d'**HyperControl** de **Nordic SoftWare**, qui permet d'imprimer des cartes par sélection soit de données soit de statut de boutons comme les cases à cocher ou les boutons radio.

On peut aussi manipuler les champs dans un rapport et faire de la concaténation de champs à l'impression. Le tout pouvant être prévisualisé. En outre le programme permet de créer des champs calculés dans un rapport. Lequel peut être éventuellement sauvegardé en format texte.

L'autre produit est **HyperBook Maker** de **Ideaform Inc.** qui autorise l'impression des cartes d'une pile sous une forme prête à relier. On peut ainsi tirer d'une pile un véritable livre qu'il n'y plus qu'à agrafer, le programme se chargeant des rectos/versos et de la numérotation. Le programme gère aussi bien une ImageWriter, qu'une Laser.

Les plus de ce mois

Trois inits, un accessoire de bureau

Tout d'abord le feuilleton **Boomerang** continue. Ce fabuleux utilitaire sous forme de CEDV apporte avec cette nouvelle version la mise des paramètres sous forme d'un fichier externe. Utile quand on passe à une version supérieure pour ne pas avoir à ressaisir toutes les listes. Pour ce qui est du fonctionnement de ce programme voir les numéros précédents de Pom's.

Avec toutes ces Inits que je vous propose tous les mois, il peut vous paraître utile de disposer d'un programme qui permet de les gérer. C'est chose faite avec **InitManager**, qui permet d'activer/désactiver une Init au démarrage et de mettre ces utilitaire dans un dossier propre. Compatible tous Mac.

Autre CEDV, un économiseur d'écran. Møire permet d'éteindre l'écran de votre Mac si vous n'y travaillez pas sans perdre son contenu évidemment. Interface très agréable, et fonctionnalités importantes.

Et enfin **Clipboard Magician**. Ce DA est un éditeur extrêmement puissant de presse-papiers. On peut avec copier dans un format, coller dans un autre, en ayant même modifié des données au passage. Étonnant.

Et pour finir, je vais vous livrer un secret : une des sources qui me permet de répondre à toute question sur tout ce qui se fait sur Mac aux États-Unis, c'est **The Macintosh Buyer Guide**. Cette revue trimestrielle compile toutes les adresses, téléphone, fax, des éditeurs américains, avec un classement par produit. Quasiment indispensable pour acheter aux USA. Ce qui souvent permet de faire de substantielles économies. J'en profite pour vous le rappeler, acheter par correspondance là-bas est d'une simplicité enfantine. Avec une carte Visa, en donnant le numéro et la date de fin de validité, par téléphone, fax ou par lettre simple, une semaine parfois moins, suffit pour voir arriver le colis convoité. Avec au bout du compte 30 à 50% d'économies. Cette revue est diffusée en France par l'A.I.C.

Adresses

Advance Gravis
6894 Palm Avenue Burnaby B.C
Canada V5J 4M3

MicroTouch Systems Inc
55 Jonspin Road
Wilmington, MA 01887, USA

TPS Electronics
4047 Transport Paolo Alto
CA 94303, USA

RasterOps
2500 Walsh Road, Santa Clara
California 95051, USA

P-Ingénierie
10 rue Mercœur, 75011 Paris

Trendware Corp
PO Box 2285 Huntington
CT 06484, USA

Millienum Software
1970 S. Coast HWY,
Laguna Beach, CA 92651, USA

Somak Software Inc
535 Encinitas Bld, Suite 113
Encinitas, CA 92024, USA

Icom Simulations Inc
648 S. Wheeling Road, Wheeling
IL 60090, USA

HJC Software.Inc
PO Box 51816 Durham
NC 27717, USA

Odesta Corp
4084 Commercial Avenue
Northbrook, IL 60062, USA

A.I.C.
54 rue Lamartine, 75009 Paris
☎ (1) 42 81 21 93

Mainstay
71 rue des Atrébates
1040 Bruxelles, Belgique

MacSELL
193 rue Jean Lolive
93500 Pantin
☎ (1) 48 43 60 33

Vous avez dit forets ?

La pince à tiercé vous vous souvenez ? Avec les disquettes de l'Apple II les petits malins se sont vite aperçus qu'il suffisait de trouer le bord de la disquette de manière symétrique au trou d'écriture, pour pouvoir écrire sur la deuxième face.

De 143Ko on passait ainsi à 286Ko pour le même prix même si ce n'était pas en continu. Le temps du bidouillage est révolu et les disquettes de 800Ko du Mac peuvent être formatées en 1,44 Méga dans les nouveaux lecteurs FDHD.

Il faut percer avec une perceuse et un petit foret, un trou symétrique au premier trou de verrouillage des disquettes. Les perceuses pour maquettes sont idéales. Une disquette ainsi manipulée, insérée dans un lecteur 1,44 Méga se voit d'office formatée dans cette capacité. Le processus ayant un système de vérification du formatage, la disquette est éjectée sans pitié si le formatage ne réussit pas. Auquel cas un petit scotch sur le nouveau trou la fait repasser en 800Ko. 80% des disquettes double face peuvent sans problème passer l'épreuve. Vous pouvez toujours essayer d'en faire autant avec vos premières disquettes de 400Ko, dont personne ne doute vous les avez quasiment toutes reformatées en 800Ko. Le taux de réussite est plus réduit mais ça marche aussi.

À vos forets.

Apple // for ever

Jean-Yves Bourdin

Nouvelles du front

Les programmeurs s'organisent

L'événement le plus intéressant pour l'Apple //, c'est sans doute la conférence des développeurs Apple // qui s'est tenue à Kansas City, à l'initiative d'A2 Central, mais avec une forte participation d'Apple.

Si je dis à l'initiative d'A2 Central, et pas de l'APDA Apple, c'est que les développeurs Apple // ont senti le besoin de s'organiser indépendamment d'Apple, à cause de la politique de marketing aberrante de celui-ci concernant l'Apple // (voir ci-dessous "Vendre des GS").

Vous trouverez nombre d'informations venant de cette conférence dans ces lignes, mais pour avoir plus de détails sur les discussions de cette conférence, il en existe un enregistrement sur cassettes audio à commander à A2 Central.

Beagle y a dévoilé comment écrire vos propres TimeOut, mais a fait promettre à chacun de ne rien révéler. Apparemment, nul ne savait à Kansas City que des sources d'applications TimeOut étaient déjà publiés, à plusieurs reprises, dans nos colonnes... Qui peut croire sérieusement qu'on peut enseigner à utiliser Sourceror sur AppleWorks et éviter qu'on l'utilise sur TimeOut?

Un des actes essentiels de cette conférence a été la constitution d'une Association permanente des développeurs sur Apple //, totalement indépendante de l'APDA et d'Apple. Sa première réunion officielle se tiendra à l'Expo Apple // de septembre à San Francisco.

Son bureau provisoire indique assez quelle sera son orientation : Barney Stone (DB Master), Roger Wagner (HyperStudio et tous les autres), Mark Simonsen de Beagle, Ron Lewin (Micol Basic), Bill Mensch (concepteur du 65816), et Tom Weishaar d'A2 Central. Développeurs Apple //, je ne peux que vous inciter très fortement à y adhérer.

Parallèlement, et toujours indépendamment d'Apple (et de Claris), s'est formée une Association des Programmeurs en... AppleWorks ! La combinaison AppleWorks 3.0/ UltraMacros 3.0 permet à AppleWorks de devenir un vrai langage, aux développements illimités. Ne serait-ce que pour rendre compatibles entre elles toutes les applications qui se greffent sur AppleWorks et/ou travaillent les fichiers AppleWorks, une telle association était bien une nécessité.

Le nouveau GS

À ceux qui croyaient que je diffusais des rumeurs incontrôlées, Apple USA a répondu le 15 août dernier : il y a bel et bien un nouveau GS. À partir du 15 août, les GS vendus aux USA ont une carte d'extension mémoire d'un Méga (comme en France) et une nouvelle carte-mère. Ils sont livrés avec le nouveau système 5.0.

La nouvelle carte-mère inclut plus d'outils en ROM (256Ko au lieu de 128Ko de ROMs), diverses améliorations dans la gestion du clavier, des améliorations pour le graphisme et le son, etc.

Finalement, le plus intéressant dans tout ça, c'est que des possibilités supplémentaires existent pour conserver les capacités des "slots internes" du GS, tout en ayant configuré le tableau de bord sur *ma carte*. Cela est possible sur l'ancien GS pour les slots 3 et 4 (techniquement : le signal *Device Select* est toujours présent), mais cela est étendu à d'autres slots dans le nouveau. C'est fort précieux : cela revient en fait à dire que nous avons plus de slots.

Apple USA a annoncé qu'il n'y aurait pas de programme de mise à niveau : les possibilités nouvelles apportées par ce GS sont minimes, comparées à ce qu'une mise à niveau coûterait. Effectivement, dans la mesure où le Méga de RAM supplémentaire n'a finalement pas été mis sur la carte-mère elle-même, sachant que le système 5.0 tourne aussi sur les anciens GS, et inclut en soft 90% des améliorations du nouveau GS, la mise à niveau ne s'impose pas.

Nouvelles d'Apple France

Trop de nouvelles des Ulis dans ce numéro. Je les résumerai donc en un seul conseil : allez à l'Expo ! Vous y trouverez un stand bien visible de l'Apple // Service Team, avec plein de nouvelles : une nouvelle édition du Guide de l'Apple //, qui a doublé de volume (de 80 à 160 pages). Je ne vous détaille pas son contenu, vu son prix...

Une démonstration des possibilités de la combinaison système 5.0/GS/Mac, et des échanges entre Mac et Apple // vous attendra au stand Apple //. Un **GSForth**, réalisé par **Alexandre Ferret** (d'Apple France), vous montrera les possibilités du 5.0. Et un jour, ce Forth vous sera vendu... au même prix que le Guide de l'Apple // (gratuit !). Seront aussi en démonstration **AppleWorks GS** et **HyperStudio**. Un spécialiste du *hard* Apple // répondra à vos questions. Et j'en passe...

À propos de la version française du 5.0 : c'est pour bientôt, c'est en cours. Je vous ai déjà conseillé, je

crois, de le réserver à l'avance chez votre concessionnaire...

En ce qui concerne le nouveau GS : l'essentiel est dans le fait qu'Apple France a pris la décision claire de vendre, dès qu'il le pourra, ce nouveau GS. La francisation demande du temps. Aucune date n'est fixée, mais les plannings pointeraient plutôt vers le mois de janvier.

Pour ceux qui voudraient mettre à jour leur GS, n'essayez pas de fracasser votre carte-mère comme je l'avais suggéré : on vous en mettrait une ancienne (en bon état, mais ancienne), à la place! Et si quelques fanas souhaitaient tout de même mettre à jour leur GS, qu'ils se posent la question : êtes-vous vraiment prêts à payer plusieurs milliers de francs pour ça ?

Dernière bonne nouvelle d'Apple : il y a désormais (enfin !) un **VS-Com 1.14** qui tourne sous GS/OS. La mise à jour est gratuite : il suffit d'envoyer sa disquette originale à Apple France.

Vendre des GS

C'était la complainte à la conférence des développeurs de Kansas City (voir A2 Central de septembre), c'est la complainte au GS Club 94, c'est la complainte des revues US (voir Nibble, Sourceror Apprentice, etc.). C'est en fait le dernier reproche sérieux que nous ayons à faire à Apple, aux USA comme en France : Apple ne sait pas vendre ses GS.

Les promesses de Sculley l'an dernier étaient vraies : Apple, au plan technique, a "mis le paquet" sur le GS. Désormais, avec le système 5.0 et la nouvelle carte-mère, les problèmes techniques sont réglés pour l'essentiel. Ce qu'Apple a fait au plan technique pour le GS est très bon, et nous tirons tous, sincèrement, notre chapeau.

Cela n'a pas toujours été vrai, mais maintenant ça l'est : le GS est désormais une machine parfaitement fonctionnelle, avec un excellent système. En France, le support est désormais clairement assuré aux Ulis par l'équipe Apple //. Ne reste plus qu'une chose, en somme : en vendre.

Suite de la page 47

```
                }
                break;
                case 1:
                  {
                  DrawString("\pMajuscule ");
                  DrawChar("");
                  DrawChar(st[(i * 6) + 5 + j]);
                  DrawChar("");
                }
              }
            }
            break;
            case 3:
              {
              DrawString("\pOption - Majuscule ");
              DrawChar("");
              DrawChar(st[(i * 6) + 3 + j]);
              DrawChar("");
              DrawString("\p, puis ");
              switch (st[(i * 6) + 4 + j])
                {
                case 0:
                  {
                  DrawChar("");
                  DrawChar(st[(i * 6) + 5 + j]);
                  DrawChar("");
                }
                break;
                case 1:
                  {
                  DrawString("\pMajuscule ");
                  DrawChar("");
                  DrawChar(st[(i * 6) + 5 + j]);
                  DrawChar("");
                }
              }
            }
            break;
          }
        }
        break;
        case Repetition:
          {
          switch (st[(i * 6) + 2 + j])
            {
            case 0:
              {
              DrawChar("");
              DrawChar(st[(i * 6) + 3 + j]);
              DrawChar("");
              DrawString("\p deux fois");
            }
            break;
            case 1:
              {
              DrawString("\pMajuscule ");
              DrawChar("");
              DrawChar(st[(i * 6) + 3 + j]);
              DrawChar("");
              DrawString("\p deux fois");
            }
            break;
            case 2:
              {
              DrawString("\pOption ");
              DrawChar("");
              DrawChar(st[(i * 6) + 3 + j]);
              DrawChar("");
              DrawString("\p deux fois");
            }
            break;
            case 3:
              {
              DrawString("\pOption - Majuscule ");
              DrawChar("");
              DrawChar(st[(i * 6) + 3 + j]);
              DrawChar("");
              DrawString("\p deux fois");
            }
            break;
          }
        }
        break;
      }
      if ((j == 0) && (st[(i * 6) + 6] == v))
        {
        j = 6;
        DrawString("\p. Ou ");
        goto Boucle;
      }
  }
  SetPort(savePort);
  }

  /*****************************************/
  PoliceComplete()
  {
  GrafPtr savePort;
  static    Rect  textV  = { 9, 9, 240, 232 },
                  text   = { 12, 12, 237, 229 };

  GetPort(&savePort);
  SetPort(pFenetre);
  SpecialeOff();
  TextFont(nPolice);
  TextSize(tPolice);
  hTexte = TENew(&text, &textV);
  TEAutoView( Oui, hTexte );

  TEInsert( &lc[0], 253, hTexte);
  TESetSelect(1, 1, hTexte);
  Vue();
  Selection();
  TEActivate(hTexte);
  AfficheCode(Speciale);

  SetPort(savePort);
  }

  /*****************************************/
  Commande(c)
  char c;
  {
  switch (c)
    {
    case 'X':
    case 'x':     Couper();     break;
    case 'C':
    case 'c':     Copier();     break;
    case 'V':
    case 'v':     Coller();     break;
    default : SysBeep(10);      return;
  }
  }

  /*****************************************/
  SpecialeOff()
  {
  GrafPtr   savePort;
  Rect      r;

  GetPort(&savePort);
  SetPort(pFenetre);
  if (Speciale)
    {
    SetRect(&r, 313, 0, 384, 0);
    r.top = -7 + (16 * Speciale);
    r.bottom = r.top + 15;
    InvertRect(&r);
    Speciale = Non;
    AfficheCode(Speciale);
  }
  SetPort(savePort);
  }

  /*****************************************/
  DesSpeciale()
  {
  GrafPtr   savePort;
  Rect      r;

  GetPort(&savePort);
  SetPort(pFenetre);
  if (Speciale)
    {
    SetRect(&r, 313, 0, 384, 0);
    r.top = -7 + (16 * Speciale);
    r.bottom = r.top + 15;
    InvertRect(&r);
  }
  SetPort(savePort);
  }

  /*****************************************/
  SelSpeciale(p)
  Point  p;
  {
  GrafPtr   savePort;
  Rect      r;
  int       i;

  GetPort(&savePort);
  SetPort(pFenetre);
  SetRect(&r, 313, 0, 384, 0);
  for (i = 1 ; i < 11 ; i++)
    {
    r.top = -7 + (16 * i);
    r.bottom = r.top + 15;
    if (PtInRect(p, &r))
      {
      if (Speciale) DesSpeciale();
      Speciale = i;
      DesSpeciale();
      break;
    }
  }
  if (Speciale) AfficheCode(Speciale);
  SetPort(savePort);
  }
  /*****************************************/
```

Aux USA, en gros, Apple se contente de vendre des GS à ceux qui en demandent. En France, il se contente d'assurer le suivi pour ceux qui en ont un. Mais la première chose qu'apprend un débutant en marketing, c'est qu'il faut vendre ses produits, et pas seulement attendre qu'on vienne les acheter.

À 10 000 F HT prix public, vous avez un ordinateur Graphique Son, une machine d'aujourd'hui, 16 bits, avec l'interface utilisateur d'aujourd'hui, extensible avec ses 7 slots, dotée de 1,2 Méga de mémoire en standard, un moniteur couleur RVB excellent. Sans compter la compatibilité Apple //, et la montagne d'excellents logiciels GS.

Qu'Apple ait, dans un premier temps, cherché à vendre le GS d'abord aux propriétaires d'Apple // 8 bits, rien que de plus normal. Mais aujourd'hui, le GS couleur 1,2 Méga coûte moins cher que ce que j'ai payé mon II Plus d'occasion. Ce n'est plus un simple *upgrade* de l'Apple //. Il est temps de le vendre à tout le monde.

Ne serait-il pas temps que les publicités d'Apple, et pas seulement les sourires des professeurs compétents, fassent comprendre à ce Proviseur de Lycée si fier d'avoir pu acquérir, sur les fonds propres de son établissement, de nouveaux compatibles PC «dont trois compatibles AT», qu'il aurait pu avoir *pour le même prix* de vrais ordinateurs pour les grands ?

Ne serait-il pas temps que ses élèves travaillent sur des machines meilleures que celles qu'ils ont chez eux, avec au moins autant de couleurs, de graphisme et de son ? Je crois cependant qu'Apple commence à se rendre compte lui-même de tout cela : c'est dans le bulletin de l'APDA Apple qu'on peut lire, sous la plume de Mike Westerfield (ByteWorks), que «le marché de l'Apple IIGS est semblable au marché du Macintosh à ses débuts, quand le Macintosh avait deux ans d'âge».

Au bout de deux ans, le Mac avait réglé ses problèmes de jeunesse, était prêt à s'imposer à cause de ses qualités internes et de ses possibilités d'évolution. Seuls Apple et une élite d'utilisateurs le savaient et le disaient alors. Ceux qui n'ont pas pris ce train l'ont regretté ensuite : ceux qui ne montent pas aujourd'hui dans le train du GS le regretteront ensuite...

Chez nous, la présence de l'Apple // à l'Expo sera visible. Et j'ai appris que je m'étais un peu trompé à propos du Sicob : Bréjoux y était bien, mais Apple aussi. En fait, Apple hébergeait Bréjoux : signe qu'Apple France veut effectivement promouvoir le GS, mais sait sur qui s'appuyer pour cela. En somme, tous ceux qui soutiennent l'Apple // travaillent de plus en plus ensemble...

Si les consommateurs sont intelligents et qu'Apple fait son travail, on peut s'attendre à un "boum" sur le GS (le nouveau, avec 256Ko de ROM et le 5.0) en janvier...

Vent d'Est

À la conférence des développeurs de Kansas City, une participation remarquée : deux développeurs soviétiques (V. Fedorov et A. Surougine) d'un Institut de Moscou. L'URSS a passé un contrat pour importer pour ses écoles 100 000 ordinateurs compatibles //e de Bulgarie, où on les fabrique sous le nom de *Pravetz*.

L'Institut recherche des producteurs de logiciels qui seraient intéressés par le marché soviétique. Il a également à vendre une excellente réalisation, un analyseur spectral, instrument de mesure scientifique qu'on ne trouve jusqu'ici que sur des configurations beaucoup plus chères.

«Apple // for ever, camarades !»

AE grossit

Il est en train de se passer pour Applied Engineering ce qui s'est déjà passé pour Beagle, et auparavant pour... Apple lui-même. L'entreprise devient si grosse qu'elle ne peut plus être dirigée par son "père fondateur", du moins si celui-ci préfère la conception des produits à la gestion d'une entreprise. PDG, c'est un métier comme un autre, après tout.

Dan Pote laisse donc son poste de PDG à un nouveau venu recruté à l'extérieur, R. J. Carroll, pour diriger pour sa part le service Recherche et Développement d'AE. Qu'AE s'organise, c'était devenu inévitable. Mais l'expérience nous enseigne que le départ de Wozniak n'a pas signifié que de bonnes choses pour Apple : espérons que l'entreprise AE saura rester aussi Apple // For Ever qu'auparavant. C'est ce qu'elle dit, et je la crois.

Laser

Il y a du monde sur le marché de l'Apple // : en-dehors d'AE, dont tout le monde sait qu'il pourrait dès demain, si Apple ne faisait plus son travail, sortir des machines meilleures que celles d'Apple, **Laser Technology** (qui appelle à juste titre le //c Plus un *clône Laser*) a déjà dans ses cartons le prototype d'un vrai portable 8 bits, et un compatible GS.

Le problème qui reste à régler est celui des ROMs : l'AppleSoft de Microsoft coûtant cher, c'est Micol qui prépare un Basic pour les ROMs du futur Laser. Et si vous savez écrire des outils pour le GS qui fassent le même travail que ceux d'Apple sans les copier (c'est du travail, mais c'est possible), votre avenir est assuré auprès de Laser Technology...

Toujours Plus

Audio Animator : pourquoi je n'ai pas de Sonic Blaster

La carte stéréo/numérisation des sons **Sonic Blaster, d'AE**, est excellente, et il n'y a rien à dire contre elle (à part cet inconvénient mineur de la protection du logiciel, qui refuse de sauver les fichiers sur un GS sans la carte). Je n'ai qu'une raison pour ne pas avoir mis de Sonic Blaster dans mon GS : j'attends mieux.

Car je connais le nom de ce *mieux* : la carte **Audio Animator**, toujours d'**Applied Engineering**. Audio Animator, c'est la Sonic Blaster, plus une interface MIDI complète compatible Passport (qui laisse libre le port série du GS), plus un logiciel séquenceur complet, plus une mini-table de mixage externe.

Au moment où j'écris ces lignes, les publicités américaines nous conseillent de passer commande tout de suite : il semble cependant que cette carte ne soit pas encore immédiatement disponible (le logiciel a pris du retard, paraît-il). Tant pis, j'attendrai un peu, quitte à me contenter du *hard* d'HyperStudio.

Octophonie

J'ai une bonne raison pour être patient : j'ai un Apple IIGS.

Parce qu'enfin il faut dire les choses nettement : loin de me plaindre, comme certains, qu'Apple n'ait pas mis de sortie stéréo au GS, je l'en félicite. C'est la même chose qu'avec la sortie Pal pour la vidéo : on peut le faire, sans gros problème, et à un coût négligeable. Qui peut le plus peut le moins.

Mais le Pal des téléviseurs, à côté du moniteur RVB Apple, c'est de la gnognotte, comme dit Zazie (et que dirait-elle à propos du NTSC...). Eh bien, la stéréo, pour l'Ensoniq du GS, c'est pratiquement la même chose.

Ne vous récriez pas : j'ai entendu le GS et la Sonic Blaster sur une bonne chaîne, et c'est magnifique. Mais je le maintiens : la stéréo est une mutilation des possibilités du GS.

Le son du GS sort sur 8 canaux, oui, huit, pas deux. Le GS joue et enregistre en *octophonie* : ai-je déjà dit que c'était une magnifique machine? Et ces huit canaux, je veux pouvoir m'en servir.

On peut toujours, avec un démultiplexeur, affecter les canaux pairs à gauche et les impairs à droite, comme font les cartes stéréo : on divise par 4 au lieu de 8. Mais on divise. Apple ne fait pas cette chirurgie, il laisse l'utilisateur libre de brider lui-même son GS.

Un GS avec Audio Animator, c'est 8 pistes et 16 canaux, Ensoniq et Midi, la qualité Apple et la qualité Applied. L'Audio Animator est la dernière pièce qui manque à mon puzzle : mais c'est pour bientôt.

Vulcain

Reçu de Bréjoux la documentation de **Vulcain**, le nouveau disque dur d'AE. C'est manifestement un choix de Bréjoux : ce qui doit être en français (la documentation utilisateur) est traduit, ce qui doit rester en anglais (les appendices techniques pour bidouilleurs) reste en anglais, mais est toujours inclus (alors que les américains n'ont pas toujours droit à cette partie).

Vu, sur cette documentation, des choses très malines en matière logicielle : un CDA qui permet de reconfigurer *à la volée* le Vulcain pour choisir la partition de boot, d'interdire l'écriture sur une partition, de ranger les têtes. Pas bête, ça, surtout quand on peut avoir jusqu'à 16 partitions sur le même disque !

Même chose avec le Driver GS/OS inclus : l'effet est le même qu'avec les drivers Apple. Les ROMs de la carte, étant dans un slot, sont ralenties à 1 Mhz. Il suffit de remplacer ces ROMs par un driver en RAM, et tout est accéléré (y compris à la vitesse Transwarp si vous en avez une).

Un P8 patché est fourni : il permet de faire reconnaître à ProDOS 8 jusqu'à 4 partitions sur le même disque (si vous avez plus de 64Ko).

Avouons-le, j'ai vu aussi quelques faiblesses : ne fournir que le préhistorique Backup // d'Apple pour faire les backups du disque dur, ce n'est pas très sérieux de la part d'AE. D'autant que Back-up // ne marche pas avec les fichiers étendus du système 5.0, lequel est aux USA le système officiel du GS.

De toute façon, je conçois mal qu'AE ne fournisse pas un utilitaire capable de faire un backup physique de l'ensemble des partitions du disque dur : il faut faire des backups par fichiers de l'ensemble des systèmes présents sur le dur. Un backup en

CP/M, un autre en Dos, un autre en Pascal, un autre en ProDOS : la galère.

En fait, la solution est toute simple: cesser de galérer avec ces systèmes d'exploitation parfaitement obsolètes, tout mettre sous 5.0 et ProDOS, et utiliser ProSel-16.

Bréjoux reconnaît que la combinaison Vulcain/Transwarp/Système 5.0 rend pratiquement inutile la Ramkeeper : les temps de chargement des applications sous GS/OS étant comparables à celles du disque RAM, et la Ramkeeper étant tout compte fait assez encombrante avec sa batterie et ses trois cartes dans un seul slot, le disque dur suffit.

C'est d'ailleurs ce qui explique cette option bizarre de la carte Vulcain : on peut faire reconnaître la carte qu'on a mise dans le slot 2 ou 5, par exemple, comme si elle était dans le slot 7 (mais seulement le 7). Parbleu, mais, puisque de toute façon cela va rendre inutilisable pour autre chose le slot 7, pourquoi ne pas mettre effectivement la carte dans ce slot ? Je ne vois qu'une réponse possible : certains bourrages de cartes AE du côté du slot d'extension mémoire doivent bloquer le slot 7. C'est une bonne chose que le Vulcain les rende inutiles.

Entre le SCSI de ToolBox et le Vulcain de Bréjoux, le prix du Mo est en forte baisse sur Apple //. Que diriez-vous d'une revue dans Pom's de ces disques durs ?

En tout cas, le disque dur devient maintenant un équipement standard des GS, comme des autres machines sérieuses. Si vous trouvez que c'est encore un peu cher, songez donc à revendre votre second lecteur 3,5...

RAM-disque dur

Si vous voulez vraiment un disque dur qui soit aussi rapide que la RAM, eh bien prenez un disque dur... RAM ! C'est ce qu'a réalisé **Cirtech** : un disque SCSI en RAM sauvegardée (qui

garde les informations même ordinateur éteint). Une sorte de grosse carte Speedisk, mais présentée comme un engin SCSI. SCSI, cela veut dire chaînable sur la carte SCSI, et compatible Mac et Apple //.

Outre la vitesse, c'est un moyen pour dépasser la barrière des 8 Mégas sur GS et 4 Mégas sur Mac Plus : plus aucune portion de la mémoire adressable n'est bloquée par un RAM-disque, tout est disponible pour les applications.

En outre beaucoup de ces applications, sur Apple // comme sur Mac, utilisent justement le dur comme mémoire virtuelle. Quand ce dur est lui-même fait de RAM, autant dire qu'on augmente la RAM de la machine.

Fastmaths, suite

Reçu également de Bréjoux la documentation de la carte **Fastmaths** d'AE. N'étant pas un technicien de la chose, je ne peux qu'apporter quelques précisions. La carte sert à la fois d'accélérateur pour les opérations mathématiques et apporte de la précision en plus (64 bits). On peut utiliser la carte dans un slot du GS, même non déclarée dans le tableau de bord, à condition de ne pas utiliser la ROM de la carte, mais le logiciel fourni sur disque.

Ce logiciel consiste en un gros ajout à Basic.System, et en des sources Merlin/APW. Cela confirme l'orientation *programmeurs*, et non *utilisateurs* de la carte : les softs fournis ont pour auteur Bob Sander-Cederlof (mais pourquoi AE ne le dit-il pas ?), ce qui est un gage de qualité.

Mais il y manque un outil SANE pour remplacer celui de GS/OS, et des patches au tableur d'AppleWorks. Si vous calculez beaucoup, ou que vous faites beaucoup de graphisme, et que vous savez programmer, la Fastmath est pour vous. Mais si vous voulez seulement accélérer le GS, c'est la TransWarp GS qu'il vous faut.

Le mange-disques

AppleWorks 3.0 en France

C'est une lettre parfaitement officielle, du service Relations Consommateurs de Claris USA, datée du 2 août : AppleWorks 3.0 (version US) est disponible en France, à un tarif d'upgrade, chez Claris-France/ P.Ingénierie. Aux USA, l'upgrade coûte 80 dollars et est valable pour n'importe quelle version d'Apple-Works. Bon, l'expérience de Claris France vous a rendu sceptique : moi aussi, puisque j'ai préféré envoyer mon AppleWorks et 80 dollars à un de mes agents en Californie, qui l'a acheté là-bas pour moi. Mais cette lettre est claire : fournir un upgrade d'Apple- Works 3.0 est une *obligation contrac- tuelle* de P. Ingénierie, dans la mesure où il a accepté de s'appeler Claris France. Je ne peux donc qu'inciter les impatients, ceux qui ne peuvent pas attendre le résultat des tractations secrètes en cours pour une version française de la 3.0, à s'adresser à Claris.

Beagle contre Beagle ?

C'est un peu pour rire que je parlais du "trust Beagle" dans Pom's 43. Et pourtant il commence à se produire pour notre bon vieux chien ce qui se passe souvent dans les grosses machines : il se concurrence lui-même. Déjà que **TimeOut Edifonts**, écrit par qui vous savez et diffusé par qui vous savez, concurrençait le **Beagle Font Editor** (et le surpasse, à mon avis du moins, bravo Daniel Lurot), voici que Beagle a en même temps à son catalogue l'excellent **Point to Point** de **Gary Little** (récupéré chez Pinpoint), et un autre logiciel de communications, **TimeOut Tele-Comm**, qui fait en gros la même chose, mais sous TimeOut.

TimeOut ReportWriter français, en revanche, c'est pour bientôt : on en est à la phase de *débugage* (des bugs américains d'ailleurs, il n'y a jamais de bugs français dans les TimeOut). Rendez-vous au prochain Pom's.

L'autre inusable

À quoi peut bien servir, de nos jours, un Apple // 8 bits ? En-dehors d'applications particulières (serveurs dédiés, instruments de mesure, logiciels éducatifs, etc.), la réponse globale tient en deux noms :

1° AppleWorks (avec TimeOut) ;

2° Print Shop.

Broderbund ne se satisfait pas, semble-t-il, du million et demi d'exemplaires de **Print Shop** qu'il a vendus, et en sort une nouvelle version. Nouvelle interface utilisateur, meilleure résolution des graphismes, possibilité de voir sur écran le résultat de son travail avant l'impression, de placer les dessins et textes où on veut sur la page, de mettre plusieurs dessins et plusieurs polices, bref, tout ce qui manquait au Print Shop 8 bits y est désormais inclus.

En prime, les calendriers qu'imprimait Print Shop Companion seront désormais réalisés directement par Print Shop. Le compagnon est lui-même renouvelé : en plus de l'éditeur de polices et de bordures, il inclut désormais un éditeur d'enveloppes, et des modèles complets pour réaliser, par exemple, toute une "Newsletter" sur son Apple // 8 bits : Print Shop, c'est bel et bien de la PAO.

Il n'y a toujours qu'une chose qui m'irrite, cette satanée protection contre la copie, qui donne une prime aux pirates, qui ont une version normale, déprotégée, et pénalise les utilisateurs honnêtes (payants). Il faudra qu'on m'explique un jour...

AppleSoft toujours vert

Il est toujours là, bien vivant, on ne s'en lasse pas. Ce bon vieil AppleSoft, pour tourner impeccablement, a besoin de trois compléments :

1° **Program Writer**, d'**Alan Bird**, diffusé par Beagle (et Bréjoux/ToolBox en France). Voir les Pom's précédents.

2° **ProCommand**, de **G. Bredon**, diffusé en France par Pom's/ Éditions MEV. Voir dans ce numéro.

3° La série des **ToolBox**, de **Roger Wagner** (si nos amis de ToolBox ne diffusent pas les ToolBox, qui le fera ?). Cette série des ToolBox est composée de commandes passant par l'& de l'AppleSoft, et sont écrites de façon relogeable : elles sont "collées" de façon invisible dans l'espace du programme Basic lui-même, et se déplacent avec lui de façon transparente.

Cela leur permet d'être toutes compatibles entre elles, de se loger commodément dans la mémoire. Cela leur permet aussi d'être compatibles avec les commandes de ProCommand. Avec la série des ToolBox (Video ToolBox, Chart and Graph ToolBox, Wizard's ToolBox), on peut pratiquement tout faire, les menus à la AppleWorks, un curseur-souris, des camemberts, tout ce que vous voulez.

Je croyais les ToolBox un peu poussiéreuses, mais Wagner vient de ressortir les anciennes mises à jour, et une nouvelle : la boîte à outils des trucs invisibles (**The Invisible Tricks ToolBox**). Je ne détaille pas, mais cette boîte à outils comprend plein de compléments utiles à ProCommand. Sacré AppleSoft : on n'en finira jamais de voir tout ce qu'on peut lui rajouter. Voyez le paragraphe sur "AppleSoft 16 bits" ci-dessous...

MicroDot

Et si, au lieu de rajouter des commandes à Basic.System, on se passait purement et simplement de lui, si on faisait une autre interface, plus économe en mémoire, entre l'AppleSoft et ProDOS 8 ?

C'est ce qu'en silence avaient déjà fait certains éditeurs de jeux. C'est ce que s'est dit Jerry Kindall, de **Kitchen Sink Software**, avec son **MicroDot**, interpréteur de ProDOS passant par l'esperluette &.

Si vous voulez gagner de la place mémoire, faire à peu près tout ce que fait Basic.System, et d'autres choses encore, MicroDot est pour vous. Un seul adversaire, en fait, à MicroDot (mais puissant) : la force de l'habitude...

Émulation

Après avoir fait l'éloge de la documentation française d'HyperStudio diffusée par ToolBox, je dois aussi faire l'éloge de celle de Bréjoux : elle est tout aussi bonne, et en plus Bréjoux rajoute une pile sur la CEE aux piles de démo d'HyperStudio. Chiche que ToolBox va rajouter ses piles. En tout cas : 1 partout.

En revanche, j'ai vu la documentation française de Merlin chez ToolBox, et c'est un gros morceau. Mais pas chez Bréjoux : 2 à 1.

Oui, mais Bréjoux a une documentation française de Publish-It, et pas ToolBox à ma connaissance : 2 partout.

Bon, je m'amuse un peu avec ce qui n'est même pas de la concurrence, mais une émulation bien sympathique. Nous ne sommes pas encore à nous plaindre de l'excès de vendeurs de softs Apple // en France. En tout cas, je sais qui est le gagnant dans l'affaire : vous et moi, les utilisateurs.

Bréjoux a aussi des versions françaises de **Top Draw** (excellent logiciel **StyleWare** dont j'espère bien que Beagle va faire ses choux gras), de **Graphic Writer** (qui sera sans doute très bien quand on en sera effectivement à la version 3), et de **Notes and Files** et **SoftWood GS File**. Ces deux derniers logiciels de gestion de fichiers ont eu un certain succès aux débuts du GS. Mais aujourd'hui, mis à part le fait qu'ils sont francisés, et sans doute pas trop gourmands en mémoire, ils me semblent, pour être francs, plutôt périmés, dépassés par AppleWorks GS (que vend également Bréjoux). La démonstration de SoftWood GS File contient même une version fossile du système pour GS qu'il est condamnable de faire encore circuler aujourd'hui (alors que l'application semble marcher sous 5.0).

Ce qui est intéressant dans tout ça, c'est la tendance qui se dessine : de plus en plus de documentations et de versions françaises, un marché GS qui, manifestement, est en croissance en France. En France comme aux USA, ça redémarre...

Sous le capot

Réencreur pour bricoleurs

Après diverses mentions dans Pom's, voici aujourd'hui le point sur le réencrage et les réencreurs de rubans d'imprimante. Vous trouverez dans les Pom's précédents les adresses pour acheter l'encre (pas chère), et les réencreurs tout faits si vous préférez payer que bricoler.

Premier point : réencrez. Je sais que vous êtes comme moi, et que vous faites durer vos rubans d'imprimante au maximum, quitte à imprimer en gras, à faire passer trois fois la tête d'impression, etc. Et vous avez aussi tort que moi.

D'abord, parce que la qualité de l'impression est directement fonction de la qualité de l'encre, c'est-à-dire de sa fraîcheur. La différence sur le papier entre une encre fraîche et un vieux ruban est *phénoménale*. J'ai moins envie de LaserWriter aujourd'hui que j'ai vu ce que donne un `pr#1:catalog` avec de l'encre fraîche. Après tout, une des supériorités de la Laser, c'est simplement d'alimenter en permanence en toner frais.

Attention : je dis la fraîcheur de l'encre, pas celle du ruban. Il arrive fréquemment qu'une cassette neuve donne des résultats médiocres à l'impression, parce qu'elle a traîné quelques mois en stock sur une étagère, et que l'encre a commencé à sécher. En revanche, un ruban vieux, mais récemment réencré, donne des résultats excellents.

Ensuite, réfléchissons : quand nous faisons durer un ruban, c'est entre autres par esprit d'économie. Or, que faisons-nous ? Nous remplaçons l'encre fraîche, qui ne coûte pratiquement rien (l'encre n'est pas chère, et il en faut très peu), par l'usure du ruban, sur lequel nous repassons plusieurs fois. Et surtout,

nous usons les aiguilles de la tête d'impression, et le mécanisme de l'imprimante : et ça, ça coûte cher. L'économie passe pour le réencrage : plus besoin de nouveaux rubans (on peut réencrer des dizaines de fois le même ruban), et l'imprimante dure plus longtemps (parce qu'on lui demande moins, mais aussi parce que l'encre fait lubrifiant pour la tête d'impression).

Voici donc comment fabriquer votre réencreur à peu de frais : il faut avant tout un moteur lent, capable d'entraîner le ruban dans la cassette à une vitesse d'un centimètre par seconde environ, et cela dans le sens indiqué par la flèche dessinée sur la cassette. En fait, si vous avez une perceuse qui fait aussi visseuse-dévisseuse, vous avez un réencreur. Il suffit de trouver quel tournevis est capable d'entraîner la cassette par en-dessous. En-dehors de cela, une bouteille d'encre spéciale, et vous avez tout le matériel nécessaire.

Le réencrage consiste simplement à faire défiler *lentement* le ruban autour d'un tube-réservoir muni d'un *petit* trou qui dispense l'encre sur le ruban.

Le réservoir est simple à fabriquer : il suffit d'un tube genre tube d'aspirine, dont le fond sera rempli sur 2 ou 3 cm de haut par un mastic quelconque non soluble par l'encre, ou ce que vous voulez. Juste au-dessus de ce fond surélevé, vous percez sur le bord du tube un *tout petit* trou.

Autour de ce tube, placez deux rondelles de caoutchouc ou de plastique : vous réglez l'écartement de ces rondelles en les faisant coulisser sur le tube, de façon qu'elles guident votre ruban à sa largeur exacte, et vous permettent de choisir sur quelle zone du ruban vous déposerez l'encre (face au trou).

Attention, c'est ici qu'il faut être précis : regardez bien votre ruban usé. Vous devez y voir deux traînées d'usure parallèles, l'une en haut, l'autre en bas. Votre ruban doit être placé sur le tube de telle façon que la traînée du bas soit juste en face du trou, et s'encre en défilant. La cassette étant un ruban de Moebius (votre ruban n'ayant donc, contrairement aux apparences, qu'une seule face), en passant deux fois la totalité du ruban vous encrerez les deux traînées.

Si votre ruban n'a qu'une seule traînée, placez le trou au bas de la traînée, de façon à réencrage le haut de la traînée au deuxième tour.

Mettez dans le tube la valeur d'un petit dé à coudre, *pas plus*, d'encre. Faites défiler le ruban. Regardez la traînée d'encre déposée sur le ruban : elle doit être continue, et large d'un millimètre au maximum. Si elle est en pointillés, le trou est trop petit ou le ruban défile trop vite. Si elle est large de plus d'un millimètre, c'est le contraire.

Laissez le ruban défiler jusqu'à ce qu'il ne sorte plus d'encre par le trou : le tour est joué. Attendez quelques heures, que l'encre se répartisse d'elle-même sur les replis du ruban dans la cassette, avant d'utiliser le ruban réencré.

Un conseil pour terminer: on met toujours *trop* d'encre. Il en faut très peu. Plutôt que de surcharger d'encre, qui ne s'enlève plus, réencrez plus souvent. Attention aussi si votre ruban interrompt à un moment son défilement sur le réencreur : le trou continuant à suinter, le ruban va être surchargé d'encre à cet endroit. En même temps que l'encre spéciale, achetez donc de l'acétone et un bon savon...

Une fois compris le principe, vous avez aussi compris qu'avec des rubans vierges, des encres de couleur, et des tubes d'aspirine, vous pouvez faire des rubans de la couleur que vous voulez (y compris le blanc, le doré, etc., qui donnent des résultats magnifiques sur papier noir).

Ventilateurs

Vous connaissez mon conseil : l'unité centrale à la cave, capot ouvert, c'est le moins cher des ventilateurs. Reste que certains d'entre vous n'ont pas de cave, ou tiennent à voir leur unité centrale. Or, le rafraîchissement n'est pas un luxe : certes, le GS est conçu pour fonctionner même quand on se brûle en touchant les composants. Mais, avec 10 degrés de moins, vous aurez moins d'erreurs 911, et il durera plus longtemps. Un mot donc sur les ventilateurs.

D'abord pour les bricoleurs : un ventilateur 220V style ventilateur de Hi-Fi se trouve pour 100 à 150 F dans les magasins de fournitures pour électronique et Hi-Fi. On en trouve pour rien du tout dans les vieux amplis jetés à la ferraille. Mais il faut bricoler.

Il y a deux inconvénients des ventilateurs : ça fait du bruit et ça prend de la place sous le capot. Pour limiter le premier inconvénient, il faut un ventilateur spécialement étudié. Le plus silencieux, à ma connaissance (et le plus économe en électricité, puisqu'il est fabriqué en Écosse !), est le **Cool-It**, de **Cirtech**, alimenté sur la prise *fan* du GS (on peut contacter ToolBox, qui importe du matériel Cirtech).

Seul problème possible : le Cool-It étant interne, comme le ventilateur Apple, certaines cartes d'extension encombrantes pourront manquer de place sous le capot. Et il vaut mieux ne pas trop *tirer* sur l'alimentation du GS, si on veut pouvoir remplir ses slots.

Pour supprimer cet inconvénient, on trouve des ventilateurs externes, qui soufflent sur le GS (ou plutôt aspirent) depuis le couvercle de son capot (lequel est plein de trous, l'aviez-vous remarqué ?), et sont alimentés directement par EDF. Leur avantage est qu'ils peuvent faire plus que simplement ventilateurs : entre autres, ils font régulateurs de tension (ce qui n'est pas inutile du tout).

Ils permettent d'avoir plusieurs sorties électriques (régulées) pour le moniteur, le disque dur, l'imprimante, le CD-Rom et le reste, toutes commandées par des interrupteurs en façade du GS (enfin). On peut donc tout allumer et tout éteindre d'un seul geste.

Enfin, ils donnent une place bien pratique pour mettre les lecteurs 3,5", qui sans cela sont généralement à côté de l'unité centrale. Si vous n'avez qu'un lecteur 3,5", on peut mettre des disquettes dans le trou restant à côté, c'est prévu. Mais je ne le conseille

pas : à l'intersection des champs magnétiques émis par l'unité centrale, le moniteur et le lecteur de disques, la boussole s'affole, et les disquettes risquent de perdre le Nord.

Les deux variantes que je connais de ces ventilateurs/régulateurs/interrupteurs, sont d'une part le classique **System Saver GS**, de **Kensington**, d'autre part le tout nouveau Conserver, d'AE. AE a un avantage : avec Bréjoux, il a un concessionnaire en France. On peut donc espérer un Conserver en 220 volts, avec des prises trois broches européennes en entrée et en sortie.

Changer la pile

Ça y est, cette irritante question des piles du GS qui s'usent trop vite est réglée. Bon sûr, mais c'est bien sang : si la pile est usée, il n'y a qu'à la changer pour une neuve ! Il n'y a pas besoin de dessouder, ça prend cinq minutes et ça coûte 10 dollars. Il suffit de commander (d'avance, de préférence) une pile de rechange (pile 3 volts au lithium) à **Nite Owl Productions**.
Le produit s'appelle **Slide-On // GS Replacement Battery**. Une remarque quand même : ne mettez pas l'ancienne pile à la poubelle, elle est très polluante. Apportez-la avec un beau sourire à votre concessionnaire...

Patchworks

AppleWorks 3.0 : c'est reparti...

Un nouvel AppleWorks, cela veut dire de nouveaux patches (même si beaucoup de patches anciens sont devenus inutiles avec la nouvelle version). Voici celui que j'ai cuisiné aussitôt après l'avoir reçu : il remplace ces satanés "ù" un peu partout par des "!" (pour les caractères souris, on verra ensuite : allez, Éric, les vacances sont finies).

```
$BLOAD APLWORKS.SYSTEM,A$2000,
    T$FF
$POKE 4335,33
$BSAVE APLWORKS.SYSTEM,A$2000,
    T$FF
```

Pour les patches UltraMacros : il faut faire `poke $10EF, $21` (au lieu de $7C).
Pour les patcheurs fous : notez que la version 3.0 semble avoir enfin utilisé les tables de datas (entre $1000 et $1100 ici) plutôt que le codage *hard* des versions précédentes.

Fichiers spartiates

Lancez GS Paint (ou PaintWorks Gold) sous le système GS/OS 4.0 ou 5.0. Faites un dessin tout bête : un petit trait noir sur un fond blanc. Sauvez avec la commande *Sauver sous* ou *Save As* (pas la commande *PO-S*) en *format écran* (pas en format GS Paint ou en format Apple), sous le nom de BLANC. Ensuite, avec *PO-O*, rechargez le dessin depuis le disque.

— Bon, ça marche, et alors?

— Eh bien, maintenant faites la même chose, tout aussi bête, avec un petit trait *blanc* sur un fond *noir* : ça ne marche plus ! Le dessin a apparemment été sauvegardé sur le disque, mais ensuite le fichier correspondant (NOIR) n'apparaît plus dans le catalogue. Plus moyen de le recharger du disque, de l'ouvrir !

Encore plus fort : quittez GS Paint, et cataloguez le disque avec vos deux dessins. BLANC fait 65 blocs, ce qui est normal, puisque sa longueur est de 32768 ($8000) octets. Mais NOIR, avec la même longueur, n'occupe que 10 blocs! Comment est-il possible de caser 32768 octets en 10 blocs de 512 octets ?
L'explication de ce mystère, c'est ce qu'Apple appelle les *sparse files*, ce que je traduis par fichiers *spartiates*. Schématiquement, quand ProDOS (8 ou 16) ou GS/OS rencontrent, durant la sauvegarde d'un fichier, un bloc de 512 octets entièrement composé de zéros, ils ne sauvent pas ce bloc sur le disque, pour gagner de la place. La longueur du fichier indiquée dans le catalogue est supérieure au nombre de blocs alloués, mais ça ne fait rien, le système chargera quand même le fichier sans problème.

On s'était très peu rendu compte de cela jusqu'ici, parce que ProDOS 8 et ProDOS 16 ne font cela qu'exceptionnellement, sur certains types de fichiers textes, très peu utilisés. Mais GS/OS le fait sur *tous* les types de fichiers, y compris les fichiers d'écrans graphiques ($C1).
GS Paint ou PaintWorks Gold reconnaissent ces fichiers à leur type ($C1 ou BIN) et à leur nombre de blocs occupés (65). Mais, si un dessin blanc (composé de $FF), occupe bien 65 blocs sur le disque, un dessin noir (au moins avec les palettes standard), est essentiellement composé de zéros : le fichier est réduit à la portion congrue, et GS Paint ne le reconnaît plus !

Cela implique une morale pour les programmeurs : il ne faut pas analyser un fichier dans un catalogue par le nombre de blocs qu'il occupe, cette pratique étant une survivance du Jurassique moyen (sous Dos 3.3, on ne pouvait guère reconnaître un dessin HGR dans le fouillis des fichiers BIN que parce qu'il occupait 33 ou 34 secteurs).
Sous ProDOS, on ne doit s'occuper que du type principal et du type auxiliaire du fichier (à la rigueur, on pourrait s'en tirer avec la longueur du fichier, mais même cela n'est pas recommandé). Encore une fois, il faut se conformer aux directives données (de longue date en l'occurrence) par le Bureau Politique.
Pour l'utilisateur, en revanche, pas d'ennuis : seules les applications qui font l'erreur de GS Paint poseront problème. Mais ProDOS 8 et ProDOS 16 chargent parfaitement en mémoire tous ces fichiers spartiates, même s'ils ne les sauvegardent pas de cette façon.
La solution est donc toute trouvée : il suffit de faire un BLOAD du fichier sous ProDOS 8, puis un BSAVE par-dessus, et NOIR retrouve ses 65 blocs. GS Paint acceptera à nouveau de le charger. Pour éviter qu'il ne le sauve à nouveau, sous GS/OS, en fichier spartiate, il suffit de le sauver

en format compressé (Paint ou Apple), la compression préalable évitant les blocs de 512 octets de zéros.

Le petit programme ci-dessous se charge automatiquement de rétablir la longueur des fichiers $C1 : je préfère cette solution à celle de patcher les applications. Songez à mettre le nom de catalogue adéquat dans la variable DI$ en ligne 1000.

```
1000 DIM A$(15) : FI = 0 : K =
   1 : D$ = CHR$ (4) : DI$ =
   "/MON.DISQUE" : PRINT D$"P
   REFIX"DI$
1010 PRINT D$"OPEN"DI$;",TDIR"
   : PRINT D$"READ"DI$ : INPUT
    A$,A$,A$
1020 INPUT A$ : IF A$ = "" THE
   N INPUT "";A$ : GOTO 1050
1030 IF MID$ (A$,18,3) = "$C1
   " THEN FI = FI + 1 : A$(FI)
   = MID$ (A$,2,16)
1040 GOTO 1020
1050 PRINT D$"CLOSE" : TEXT :H
   OME : IF FI = 0 THEN 1070
1055 PRINT D$"CAT
1060 FOR I = K TO FI : INVERSE
   : PRINT A$(I) : NORMAL : PR
   INT D$"BLOAD"A$(I);",A$1000
   ,T$C1 " : PRINT D$"UNLOCK"A
   $(I) : PRINT D$"BSAVE"A$(I)
   ",A$1000,L$8000,T$C1" : K =
   K + 1 : NEXT
1070 PRINT D$"CAT" : END
```

Encore une partie et j'éteins

Mieux que sur Mac II

Une étiquette sur la boîte le proclame : la version IIGS est plus rapide que la version Mac II couleurs ! Je ne sais pas si c'est vrai, mais si ça l'est, ce n'est pas forcément un avantage : j'ai fini par piquer une colère contre la Transwarp, qui accélère les choses sans se faire remarquer, et me faisait faire des scores ridicules !

Ce jeu s'appelle **Crystal Quest** et est édité par **Casady and Greene**. Le principe en est simple : il s'agit simplement d'animation de *shapes*, il faut manger tout ce qui rapporte des points avant de se faire manger. Ce n'est pas un jeu trop intellectuel, mais on s'y accroche bougrement. D'autant que la "bande-son" est vraiment, vraiment, très drôle.

Bref, je vais vous faire un aveu : ce jeu est sur mon disque dur...

Allô, Michel...

— Ah, bonjour, Jean-Yves. Tu remercieras les lecteurs de Pom's de ma part : avec tout le lunarium qu'ils m'ont envoyé, ils ont permis de sauver la démocratie sur cette planète.

Pour leur montrer ma reconnaissance, je leur donnerai un conseil : dans **Test Drive** //, d'**Accolade**, on fait de meilleurs scores sur le circuit de la disquette "California Challenge", moins difficile que les montagnes de la disquette Master. Et je vais te confier quelque chose : je suis sûr que quand ils mettront une Chrysler dans les disquettes de Supercars, je battrai tout le monde...

— Eh, mais Michel, tu causes, tu causes : tu ne joues pas?

— Si, bien sûr. Simplement, là, j'ai le temps : je crée. Je fabrique mes propres niveaux pour **Arkanoid 2**, de **Taito**. Car le nouvel Arkanoid a enfin ce que nous avons tous demandé : un éditeur de tableaux. Éditeur simple et impeccable, mais tu peux fabriquer des niveaux superbes. Et, tu comprends, avec les nouvelles caractéristiques assez démentes du jeu, je trouve que le hasard tient un tout petit peu trop de place dans le jeu au détriment de l'habileté : quand tu as tous ces bazars qui tombent et 50 balles qui partent, le jeu joue tout seul en fait. Et puis, j'aimerais savoir si on peut gagner sur un niveau avec trois rangées de briques qui se reconstituent. Bon, je te laisse, il faut que je l'essaie. Si tu veux, je t'enverrai mes niveaux quand j'aurai fini de les tester.

— Merci, Michel. Peut-être un jour, qui sait, une DPA de niveaux pour Arkanoid 2 ?

18 trous

Le golf devient un sport à la mode, et se démocratise. Après Mean 18, c'est tout simplement **Jack Niklaus**, un champion américain, qui vous initie aux joies des 18 trous chez **Accolade**.

Un amateur consulté m'a affirmé que la simulation était excellente, meilleure que Mean 18. Dommage simplement qu'on ne puisse pas créer ses niveaux, comme dans Mean 18.

Pour s'amuser

Britannica Software est une maison qui s'impose sur le GS : c'est elle entre autres qui diffuse les jeux de "Fanfare", un nouvel éditeur. Outre **Laserforce**, dont je parlerai plus longuement dans le prochain Pom's, deux nouveaux jeux **Fanfare** m'ont plu ce mois-ci :

— **Gnarly Golf** est un mini-golf moitié billard très amusant, dans le genre Zany Golf (mais pas tout-à-fait aussi drôle, à mon avis) ;

— **The Great Western Shootout** est un jeu de tir amusant, mais difficile dès qu'on avance un peu.

Sans prétentions, ces jeux font tout de même passer un bon moment sur le GS.

16 bits

76 et 89

C'est quand même un peu fort qu'en cette année de Bicentenaire, j'aie à vous entretenir d'un excellent logiciel de simulation, extrêmement intéressant et bien fait, un de ces éducatifs auxquels on joue pour le plaisir, logiciel de simulation de la Révolution... américaine de 1776 !

Éditeurs de logiciels, à quoi servez-vous? Et Ministère de la Culture, à quoi sers-tu ?

Bon, c'était ma colère de ce numéro. Reste à vous dire un mot de l'excellent **Revolution 76**, de

Britannica Software, l'éditeur de logiciels qui monte sur GS. En gros, je dirais ceci : comme cours d'anglais, comme cours d'histoire des USA en même temps, c'est passionnant. Sûr que les élèves voudront rester après la sonnerie.

5.0 : les bonnes surprises

- Vous connaissiez déjà, avec le 4.0, le truc d'appuyer sur Escape au début du chargement. Avec le 5.0, c'est toujours vrai, mais il y en a un autre : appuyez sur les deux ...

- Dans le Finder, appuyez sur *shift* et *option*, et cliquez sur la pomme de la barre de menus. Choisissez la nouvelle option *About*, cliquez ensuite sur chacune des icônes de la fenêtre...

Le sens de l'humour ne se perd pas, chez Apple...

5.0 : les mauvaises surprises

- Apple a lui-même rendu public un bug qu'il a laissé dans la version 1.3 de Basic.System. Je ne détaille pas ce bug, mais il peut altérer des fichiers sur disque. Conclusion : recopiez le Basic.System du 4.0 sur votre copie du 5.0, en attendant la correction.

- Si vous allez jusqu'au bout des options du tableau de bord "de bureau" (en NDA, à la Mac) du 5.0, vous trouverez une option de reconfiguration du langage des applications QuickDraw (AppleWorks GS etc.). Je conseille très vivement de choisir l'option *pas de reconfiguration*, et de continuer à utiliser le NDA **Clavier bien tempéré d'Yvan Koenig** et autres, dans sa version la plus récente.

L'option du tableau de bord de GS/OS est compliquée à utiliser, incomplète, gène certaines applications. Le Clavier bien tempéré est complet, clair, bien documenté, et bien débugué (et rajoute des possibilités bien pratiques de sauvegarde et d'impression de l'écran graphique, d'éjection, etc.).

Messieurs d'Apple USA, pourquoi ne pas demander à des français de faire cette option eux-mêmes ? D'autant que j'ai vu, grâce à la seconde bonne surprise, un certain Xavier Schott dans l'équipe des auteurs du 5.0...

- Je crois bien avoir rencontré un bug assez grave dans le fonctionnement du Standard File (outil 23). Quand, en utilisant l'option *Sauver sous* ou *Save as* du menu-Fichiers (File) des applications de bureau, on veut sauver *sous le même nom* un fichier déjà existant dans le catalogue concerné (par exemple pour sauver une image sous un autre format, ce qui ne peut pas être fait avec PO-S), on devrait avoir une fenêtre de dialogue à boutons nous demandant confirmation.

Avec le Standard File du 5.0, on a au mieux rien du tout (pas de dialogue, et pas moyen non plus de sauver le fichier, à moins de changer de catalogue ou de nom), et au pire un beau plantage (le fichier est perdu). Comme cela arrive aussi bien avec Arkanoid 2 qu'avec PaintWorks Gold ou AppleWorks GS, je pense que c'est bien un bug du système.

En tout cas, j'ai renommé le fichier `/SYSTEM/TOOLS/TOOL023` du 5.0 en `/SYSTEM/TOOLS/TOOL023.5.0` et j'ai rajouté dans `/SYSTEM/TOOLS` du 5.0 le TOOL023 du 4.0 (cette modification peut être défaite et refaite à volonté, entre deux applications).

- Il semble que dans certaines configurations du GS et d'AppleWorks avec TimeOut, UltraMacros, et un assortiment de patches variés, AppleWorks 8 bits plante quelquefois au chargement quand on le lance depuis le 5.0 (sans doute le retour d'une ancienne erreur de ProDOS 16 signalée dans nos colonnes par Pierre Demblon). Le remède : celui de Pierre Demblon, un petit Reset préalable, sous ProSel 8 ou dans un mini-fichier pour lancer AppleWorks.

- J'ai retrouvé dans une documentation de GS/OS comment maîtriser la numérotation des *devices* GS/OS, qui m'a fait grincer des dents avec ProSel 16 (voir Pom's 43) : elle dépend tout simplement de l'ordre des fichiers de drivers dans le sous-catalogue

`/SYSTEM/DRIVERS`. Sachant que le disque de boot sera toujours placé en n° 1, si vous mettez le driver `SEXY.DRIVER` en tête du catalogue, votre disque dur et votre lecteur de CD-Rom seront placés en 2 et 3, le driver suivant affectera ses lecteurs en 4, etc. En jonglant un peu, je suis parvenu à placer quelques-uns de mes disques à un numéro correspondant à celui du slot dans lequel ils sont...

Softs Wagner : quelques problèmes

Quelques problèmes encore avec les logiciels de Roger Wagner, dont les utilisateurs doivent être avertis.

- **SoftSwitch** ne fonctionne pas avec le système 5.0 : on ne peut pas sauver un espace de travail sous P8 et Basic.System, Softswitch croyant qu'il s'agit d'une application ProDOS 16.

- **MacroMate** ne fonctionne pas encore impeccablement en clavier français : essayez, sous Basic.System, d'enregistrer et de renvoyer la macro `CALL - 151`. Les touches lettres et chiffres (minuscules et majuscules) du clavier fonctionnent bien, mais pas la plupart des autres.

- Avec le 5.0 et quelques accessoires, **The Graphic Exchange** ne trouve pas libres les bancs qu'il souhaite en bas de la mémoire, et refuse de se charger.

Soyez tranquille cependant : Roger Wagner est au courant de tout ça, et y travaille.

- Enfin je rappelle à ceux qui ont été surpris par HyperStudio que ce n'est pas encore une version définitive, que la mise à jour sera gratuite. Et que l'interface "à la Mac" ne doit pas nous cacher la nécessité de... lire les documentations. L'hypermédia, ça doit s'apprendre, comme le reste.

Éditeurs de ressources

Ce n'est pas encore ResEdit du Mac, mais ça vient. Désormais, pour programmer fenêtres, menus, boutons, contrôles et Cie sur le GS, nous allons simplement dessiner le truc à l'écran, et cocher des cases, au lieu de ces immenses tableaux de datas à remplir bit par bit qu'il fallait jusqu'ici écrire, quel que soit le langage de programmation utilisé.

Le seul à ma connaissance à intégrer totalement la notion d'éditeur de ressources au sens du système 5.0 (c'est-à-dire au sens du Mac), à faire donc des fichiers objets avec ressources, c'est **TML Pascal //**, de **TML Systems**. TML Pascal étant déjà, pour le moment, le langage le plus utilisé sur GS, il va évidemment faire un tabac.

Mais bien entendu Apple, avec APW, et ByteWorks, avec Orca, sont sur le même coup, puisque c'est une caractéristique du système 5.0.

D'ores et déjà, même si elles ne font pas des fichiers avec ressources, nombre d'applications nous permettent de dessiner directement sur l'écran ce que nous voulons, l'application se chargeant de transformer tout ça en code source.

Vous trouverez dans le sous-catalogue `/SUPPLEMENT` de la disquette 3,5" de ce numéro une application freeware appelée **Dialog Maker**, de **Steve Stephenson**, qui fait cela pour Merlin 16 : dessinez vos fenêtres, cliquez sur leurs caractéristiques. Quand c'est fini, sauvez le code source sur la disquette : il ne reste plus qu'à le coller dans votre source Merlin.

Plus complète est l'application **Design Master**, montrée par **Chris Haun** à Kansas City. C'est un indispensable quel que soit le langage dans lequel vous programmez. Ariel Publishing a décidé de la diffuser pour environ 30 dollars.

- Enfin, je dirais un mot ici d'un excellent ensemble de programmation en assembleur 8 bits pour ProDOS 8, apporté à Kansas City par son auteur, un employé d'Apple (Eric Soldan), qui le diffuse gratuitement. **App Builder** est un super-ensemble de macros extrêmement complet, en fait un langage. Pour le moment, seule existe la version pour MPW du Mac, mais bientôt les versions APW et Merlin.

AppleSoft 16 bits

Rien à faire : même si le GS n'est pas prévu pour cela, nul n'empêchera les utilisateurs de chercher à utiliser ses capacités graphiques et sonores avec l'AppleSoft. Voyez la collection de Pom's, par exemple.

Une des premières tentatives globales pour permettre de gérer le GS à partir de l'AppleSoft fut Iconix et Sonix, de So What Software. Ça marche, surtout Sonix, mais ça fait des programmes en Poke, Peek et Call... totalement illisibles.

Lynx Computer, une nouvelle entreprise domiciliée (est-ce un hasard ?) à quelques encablures d'Applied Ingenuity, nous propose une formule plus claire : utiliser les &, elle aussi, pour gérer le graphisme du GS avec **SuperGraphix 256**, et la musique avec **SuperMusic**. `& DRAW`, `& PLAY`, `& COLOR`, `& WAVE`, je comprends.

Deux choses fort intéressantes dans SuperGraphix : d'une part la possibilité d'afficher 256 couleurs en même temps, et de mélanger mode 640 et mode 320 sur le même écran, qui est dans le GS, est directement offerte aux programmeurs. D'autre part, la possibilité d'animer des formes (shapes) en SHGR, formes qu'on aura découpées dans n'importe quelle image SHGR des DPA Pom's, avec le programme (shape grabber) fourni.

SuperMusic, quant à lui, nous rappelle que l'Ensoniq a 32 oscillateurs indépendants, dont 30 utilisables : avec SuperMusic, on peut faire jouer les 30 en même temps

À lire

Au boulot

Même si Apple ne le sait pas assez, un Apple // est aussi une excellente machine pour travailler. C'est ce que démontre très bien la nouvelle revue **// At Work : Apple // Productivity News.**

Barney Stone, de Stone Edge Technology, qui produit une des meilleures bases de données tournant sur ordinateur personnel (DB Master), en est l'éditeur. On trouve dans cette revue ce qu'on ne trouve pas ailleurs, des "nouvelles d'entreprise" d'Applied Ingenuity, Berkeley SoftWorks, et d'autres. Les propos visant Apple y sont parfois assez durs : qui aime bien châtie bien.

800 pages de codes

800 pages de codes d'imprimante, ça ne se lit pas facilement dans le métro. C'est en fait un manuel de référence des codes d'imprimante, de toutes les imprimantes imaginables : bref, il ne sera guère utile qu'aux développeurs qui écrivent des drivers d'imprimante. Mais à ceux-là, il sera indispensable. Il s'appelle **The HandBook of Printer Commands**, et est disponible (en trois volumes!) chez **A2 Central**.

Bibliographie pour débutants

Plusieurs d'entre-vous me l'ont demandé : que faut-il lire quand on est un utilisateur débutant du GS ? Et que faut-il lire quand on est débutant et qu'on veut programmer son GS ?

Il m'est très difficile, je l'avoue, de répondre aux premiers : il nous faudrait un bon *Guide de l'utilisateur* de GS, en français. Mais ça n'existe pas, à ma connaissance du moins. C'est au point que j'ai songé... à en écrire un moi-même (quelque chose du genre "La Pomme sans les Pépins"). Faites-nous savoir si cela vous intéresse...

Pour les seconds, c'est plus facile : tous les titres ont déjà été mentionnés dans Pom's. La seule difficulté est de ne pas se laisser impressionner par le langage mentionné dans le titre du livre : programmer le GS en assembleur, en C, en Pascal, etc., c'est toujours en fait la même chose. C'est interfacer la Boîte à Outils du GS. Lisez donc les ouvrages vous introduisant à la ToolBox, quel que soit leur langage de référence. Pour programmer le GS, prenez le langage que vous connaissez déjà. Si vous n'en connaissez aucun, je conseillerais TML Pascal // (voir ci-dessus).

Vous trouverez les références nécessaires dans les Pom's précédents. Je conseille en premier lieu les livres d'Apple même, qui sont pédagogiquement excellents. Et vous ne pourrez pas éviter les deux volumes du ToolBox Reference.

Supplément

Sur la disquette 3,5" de Pom's 44 disponible en option, dans le sous-catalogue /SUPPLEMENT, vous trouverez quelques programmes freeware intéressants pour GS essentiellement :

- **Dialog Maker**, l'excellent utilitaire de fabrication sans peine de code source pour Merlin (voir le passage sur les éditeurs de ressources).
- **Desk Color**, un NDA amusant qui vous permet de changer à tout moment la couleur de fond du bureau.
- **Concentrate**, un jeu énervant qui nous montre combien peu nous avons de mémoire (forcément, la machine en a plus ; il faut donc la forcer à se tromper pour avoir des chances).
- **Dirmap**, un utilitaire qui nous montre de façon graphique comment sont occupés nos disques. Ce n'est pas seulement amusant, c'est utile.
- Quelques icônes pour varier l'aspect du Finder.
- Un utilitaire bien précieux pour ceux qui ont une PC Transporter: il formate automatiquement un disque ProDOS en MS-DOS-beurk. C'est particulièrement précieux avec les programmes de RAM-disque de ProSel.
- Un **Puzzle** un peu énervant aussi : après vous être échiné quelques heures, au bord de l'asphyxie, vous demandez à voir la solution. Et hop, en moins de 10 secondes, on vous montre combien c'était facile !
- Enfin **Quadomino**, un gros NDA un peu spécial : c'est le jeu Tetris présenté en accessoire de bureau !

Adresses

// At Work
P.O. Box 3200, Maple Glen, PA 19002, USA

A2-Central
P.O. Box 11250, Overland Park, Kansas 66207, USA

Accolade
20813 Stevens Creek Boulevard, Cupertino, CA 95014, USA

Apple // Developers Association
c/o Tom Weishaar, A2 Central

AppleWorks Programmer's Association
6531 Lexington Avenue, Los Angeles, CA 90038-1451, USA

Applied Engineering
P.O. Box 5100, Carrollton, TX 75011, USA

Ariel Publishing
P.O. Box 266, Unalakleet, Alaska 99684, USA

Beagle Bros
6215 Ferris Square, Suite 100, San Diego, CA 92121, USA

Bréjoux
29 Rue Montribloud, 69009 Lyon
☎ : 78 36 52 69

Britannica Software
345 Fourth Street, San Francisco, CA 94107, USA

Broderbund Software, Inc.
17 Paul Drive, San Rafael, CA 94903, USA

ByteWorks
4700 Irving Boulevard NW, Suite 207, Albuquerque, NM 87114, USA

Casady and Greene
P.O. Box 223779, Carmel, CA 93922, USA

Cirtech
Currie Road Industrial Estate, Galashiels, Selkirkshire, TD1 2BP, Scotland, Royaume-Uni

Claris France
10 rue Mercoeur, 75011 Paris
☎ : (1) 43 70 71 72

Claris Usa
5201 Patrick Henry Drive, P.O. Box 58168, Santa Clara, CA 95052-8168, USA

Kensington Microware
251 Park Avenue South, New York, NY 10010, USA

Kitchen Sink Software
903 Knebworth Ct, Westerville, OH 43081, USA

Laser Computers
550 E. Main Street, Lake Zurich, IL 60047, USA

Micol Systems
9 Lynch Road, Willowdale, Ontario, Canada M2J 2V6

Mindscape
3444 Dundee Road, Northbrook, IL 60062, USA

MMP Systems
20681 Celeste Circle, Cupertino, CA 95014, USA

Nite Owl Productions
5734 Lamar, Mission, KS 66202, USA

Roger Wagner Publishing
1050 Pioneer Way, Suite P, El Cajon, CA 92020, USA

So What Software
10221 Slater Avenue, Suite 103, Fountain Valley, CA 92708, USA

Taito
267 West Esplanade, 2nd Floor, North Vancouver, BC, Canada

TML Systems
8837-B Goodbys Executive Drive, Jacksonville, Florida 32217, USA

Toolbox
35 Bvd Richard Wallace, 92800 Puteaux
☎ : (1) 47 72 27 36

Pom's Le Tarif

* le numéro entre parenthèses indique la revue dans laquelle le produit est décrit.

Revues

n° 8 à n° 26, chacune	**40,00**
n° 27 à n° 44, chacune	**45,00**

Recueils Pom's

Ces recueils regroupent 4 numéros de la revue. (Recueil 5 à 8 épuisé)

Recueils n° 1 à 4, 9 à 12, chacun	**140,00**

Câbles communication

Pour la mise en œuvre de ces câbles, se reporter aux numéros 27, 28, 30, 31, 33 et 34 qui présentent les programmes de communication reliant ordinateur et Minitel. Préciser le ou les modèles d'ordinateur.

Câble Minitel/ordinateur	**225,00**
Câble Ordinateur/ordinateur	**225,00**

Reliures

Pour la protection et le classement de 6 numéros de Pom's (un an)

Reliure	**75,00**

Disquettes Pom's, Macintosh

Ces disquettes regroupent l'ensemble des programmes pour Macintosh d'une revue. Programmes du domaine public, polices de caractères, nouveaux systèmes, etc. y sont ajoutés.

3,5", de la n° 16 à la n° 44, chacune	**80,00**

Disquettes Pom's, Apple //

Ces disquettes regroupent l'ensemble des programmes pour Apple // d'une revue.

5,25" 140Ko, de la n° 1 à la n° 44, chacune	**80,00**
3,5" 800Ko, de la n° 29 à la n° 44, chacune	**80,00**

Logiciels pour Apple //

Ces logiciels peuvent être livrés en 140 ou 800Ko (préciser). Les logiciels américains sont accompagnés d'une documentation française.

	Tarif Joker♣	Tarif normal
ProCommand	*450,00*	**500,00**
ChessMaster GS 800Ko (39)	*630,00*	**700,00**
Arlequin 140Ko (39)	*135,00*	**150,00**
Copy][+ version 8.3 (37)	*450,00*	**500,00**
ProSel (36)	*450,00*	**500,00**
Compilateur Beagle (35)	*675,00*	**750,00**
Big U 140Ko (35)	*400,00*	**450,00**
Pom_Link 3.1 (34/35)	*400,00*	**450,00**
InterPom's 2.0 (31)	*540,00*	**600,00**
Clv_Pom's (31)	*180,00*	**200,00**
EPE 5.1 (15/23)	*180,00*	**200,00**

Logiciels pour AppleWorks

Ces logiciels dotent AppleWorks de la souris, de polices de caractères, etc.

	Tarif Joker♣	Tarif normal
MacroTools II (44)	*270,00*	**300,00**
DeskTools II (44)	*495,00*	**550,00**
ExtraFonts (43)	*450,00*	**500,00**
TimeOut PowerPack (41) / **TimeOut FileMaster (41)**	*675,00*	**750,00**
TimeOut SpreadTools (41)	*495,00*	**550,00**
MacroTools (41)	*270,00*	**300,00**
TimeOut DeskTools I (41)	*495,00*	**550,00**
TimeOut Paint(39)	*270,00*	**300,00**
TimeOut SuperFonts (38)	*720,00*	**800,00**
TimeOut UltraMacros (38)*	*585,00*	**650,00**
TimeOut Graph (38)	*765,00*	**850,00**
TimeOut SideSpread (38)	*495,00*	**550,00**
PathFinder (38)	*135,00*	**150,00**

Polices de caractères

Ces polices de caractères de type Macintosh fonctionnent avec AppleWorks & SuperFonts. Une trentaine de polices par disquette.

'Polices' 001 à 018, 140 ou 800Ko, chacune	**100,00**
Catalogue de référence Unipol	**80,00**

Shareware Apple //

Disquettes présentées dans les numéros 38, 39, 41, 42, 43 & 44.

DPA 001 à 026, 140 ou 800Ko, chacune	**100,00**

Logiciel pour IBM PC

* Accompagne le livre 'Les Secrets de Multiplan' de H. Thiriez

Les Secrets de Multiplan*		**175,00**
InterPom's	***540,00***	**600,00**

ZipChip

Nouveau processeur 4 Mhz pour les Apple / /e et //c. Description dans le numéro 39 de Pom's.

	Tarif Joker♣	Tarif normal
ZipChip 4 Mhz	***2 250,00***	**2 500,00**
Port (urgent recommandé)	***50,00***	**50,00**

Logiciels pour Macintosh

	Tarif Joker♣	Tarif normal
Sargon 4 (40)	***540,00***	**600,00**
Excel Efficace		**175,00**
MacAstuces		**200,00**
Le Livre du Macintosh		**200,00**
Copy II Mac version 7.2 (37)	***450,00***	**500,00**
Pom_Link 3.1 (34/35)	***400,00***	**450,00**
InterPom's 2.0 (31)	***540,00***	**600,00**
Clv_Pom's (31)	***180,00***	**200,00**
Raccourci (26)	***180,00***	**200,00**

Shareware Macintosh

Disquettes présentées dans les numéros 22 & 30.

Mac A, B, C, D, E, F, G, H, I, chacune	**100,00**
ADN (6 disquettes)	**300,00**

Abonnements 1 an (6 numéros)

♣L'abonnement à la revue Pom's comprend l'attribution automatique de la carte 'Joker'. Celle-ci offre un accès privilégié à une assistance téléphonique et donne droit aux remises sur les produits Pom's.

Revue seule	**225,00**
Revue et disquettes Apple // 140Ko	**625,00**
Revue et disquettes Apple // 800Ko	**625,00**
Revue et disquettes Macintosh	**625,00**

Gagnez du temps. Commandez par Minitel au :

(1) 39 53 04 40

Bon de commande

Je désire recevoir :

Quant.	Produit	Prix total

J'ai un lecteur Apple // 140Ko ❑

J'ai un lecteur Apple // 800Ko ❑

Montant _____

Port avion hors CEE

15,00 F par revue et/ou disquette _____

Montant total _____

Ma commande atteint 1400,00 F (abonné 1000,00 F), je désire recevoir le programme Arlequin qui m'est offert.

oui / non

Je règle par : Chèque

Carte Bleue/Visa/Eurocard/Mastercard

Numéro _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Expiration _ _ /_ _

Signature

Nom _____

Prénom _____

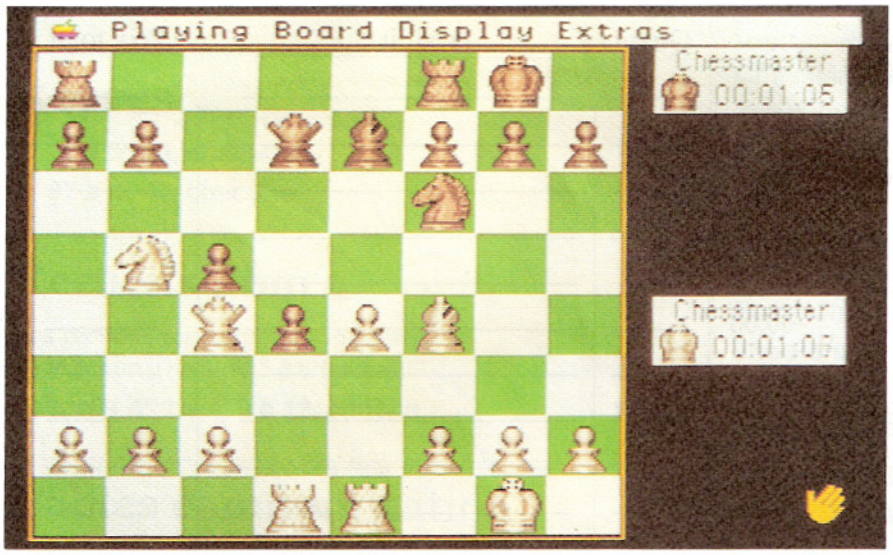
Adresse _____

_____

n° de carte Joker _ _ _ _ _ _ _ _ _

CHESSMASTER 2100

LE MAÎTRE



POM'S VOUS PROPOSE AUJOURD'HUI SUR GS UNE EXTRAORDINAIRE VERSION DU MAITRE DES JEUX, CAPABLE DE PERFORMANCES D'EXCEPTION.

NIVEAUX DE JEU

LE SOMMET DE LA REFLEXION INFORMATIQUE : CHESSMASTER 2000 + L'EQUIVALENT DE 10 ANNEES DE RECHERCHE

BIBLIOTHEQUE D'OUVERTURES CONSIDERABLE (PLUS DE 150000 COUPS)

NOMBRE ILLIMITE DE NIVEAUX DE JEU, **DE DEBUTANT A GRAND MAITRE**

POUR CHAQUE NIVEAU, OPTION 'JEU SIMPLIFIE'

EN MODE APPRENTISSAGE, CHESSMASTER 2100 INDIQUE LES COUPS ET LES PRISES POSSIBLES

TEMPS DE REFLEXION

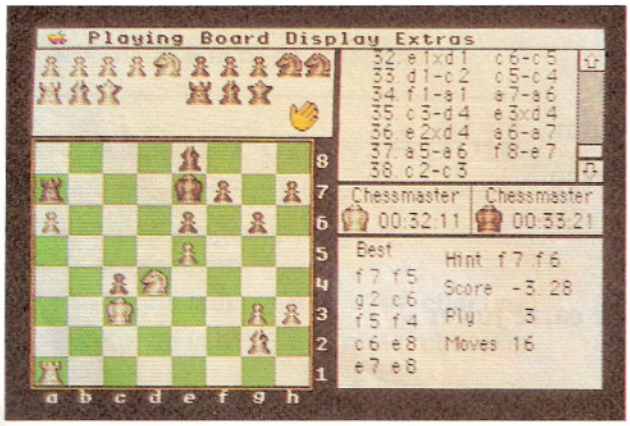
CHOIX DU TEMPS DE REFLEXION PAR COUP, PAR PARTIE, EGALEMENT EN COUPS PAR MINUTE ET **TEMPS STANDARDS DES TOURNOIS**

OPTION TEMPS EGAL, OPTION 'BLITZ'

DES AIDES

CHESSMASTER 2100 PEUT SUGGERER UN COUP, **REVENIR EN ARRIERE** ET REJOUER LES COUPS

POSSIBILITE DE CHANGER DE COTE A TOUS MOMENTS



AFFICHAGE

UTILISATION DES **RESSOURCES DU GS** : SUPER HAUTE RESOLUTION, MENUS DEROULANTS, SOURIS...

PENDULES TEMPS REEL A L'ECRAN

AFFICHAGE DE L'ECHIQUIER EN 2D OU 3D

POSSIBILITE DE DESSINER SOI-MEME LES PIECES, DE TOURNER L'ECHIQUIER PAR QUART DE TOUR, DE JOUER 'A L'AVEUGLE'

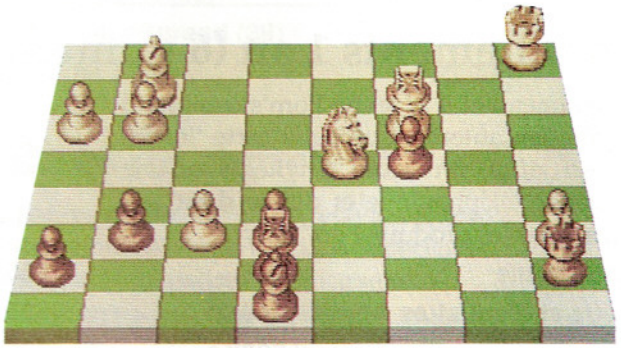
AFFICHAGE POSSIBLE DE VOS COMMENTAIRES SUR LES COUPS, DES PIECES DEJA PRISES, DE LA REFLEXION DE CHESSMASTER

POUR PROGRESSER

AFFICHAGE POSSIBLE DE LA REFLEXION DE CHESSMASTER

POSSIBILITE D'ANALYSE DES PARTIES

110 PARTIES DE MAITRES ENREGISTREES AVEC COMMENTAIRES



SAUVEGARDE ET IMPRESSION DES PARTIES

RESOLUTION DE PROBLEMES D'ECHECS

PARTIES A 'AVANTAGES'

MODE D'EMPLOI COMPLET POM'S, EN FRANCAIS, COMPRENANT UN MANUEL D'APPRENTISSAGE DE LA FEDERATION US D'ECHECS ET UN HISTORIQUE DES ECHECS PRESENTANT LES MAITRES

DE PLUS, VOTRE GS PARLE...

Éditions MEV — 12, rue d'Anjou — 78000 Versailles ✆ (1) 39 51 24 43 — Minitel (1) 39 53 04 40

La quatrième Génération

Le célèbre programme d'échecs Sargon 4 est aujourd'hui doté de l'un des algorithmes de réflexion les plus puissants.

145 000 coups d'ouverture

Il offre une bibliothèque d'ouvertures de 145 000 coups qui lui permet d'économiser un temps précieux en début de partie.

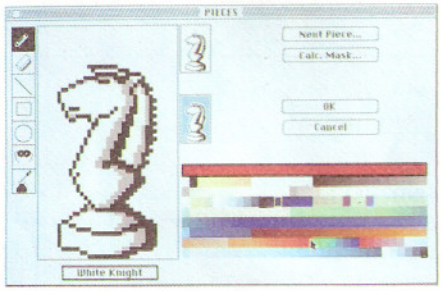
Sargon 4 joue fort bien et est à même de s'adapter à **tous les niveaux**.

À la portée du débutant

Sargon 4 pense aux novices :

Le **livret explicatif** donne les bases nécessaires. Un niveau de jeu spécial 'débutant' est prévu.

Il est en outre possible de revenir en arrière, de changer de camp, de **modifier les positions**... et même de suivre la réflexion de Sargon.



Affichage 3D

Sargon 4 présente la table d'échecs en deux ou trois dimensions (perspective) avec **cinq jeux de pièces**.

Sur Macintosh II, l'affichage est **en couleurs**, rendant ainsi l'échiquier encore plus lisible.

Des pièces éditables

Vous dessinerez vous-même vos pièces pour l'affichage en perspective. Vous pourrez aussi modifier les formes et couleurs des jeux de pièces proposés.

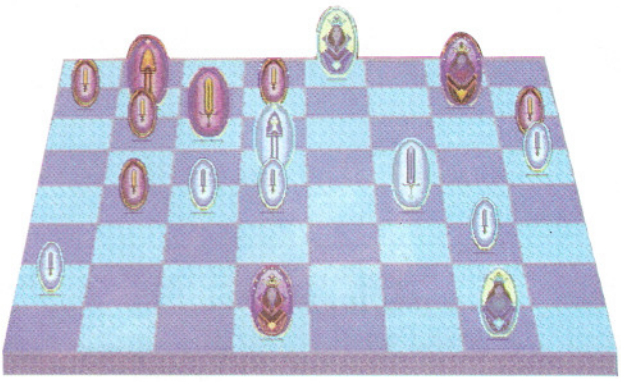
Sargon 4 sait rejouer toute une partie **en analysant chaque coup** ; cette méthode d'apprentissage est incomparable. Et, pour aller plus loin, vous étudierez les **107 parties célèbres** que Sargon vous propose.

Et de plus...

Sauvegarde, impression des parties
Pendules d'échecs à l'écran
Affichage des coups joués
Changement du sens de l'échiquier
Recherche de mats
Temps de réflexion de 5 secondes à l'infini
Parties à temps limité
Possibilité de proposer le nul...

Sargon 4 fonctionne sur Macintosh 512 avec lecteur 800Ko, Mac Plus, Mac SE et Mac II.

Guide d'apprentissage et mode d'emploi complets Traduction intégrale



Éditions MEV Revue Pom's
12, rue d'Anjou — 78000 Versailles
☎ (1) 39 51 24 43 - Minitel (1) 39 53 04 40

Essai dans le numéro 40 de Pom's
Bon de commande en fin de ce numéro

Livres

Où vous les procurer ?

Pour les documentations vous connaissiez :

- InterÉditions ;
- Prim'Vert...

Mais connaissez vous...

Infoteque
32, rue de Moscou
75008 Paris
✆ (1) 45.22.67.01

Petites annonces

Vend Unité Centrale d'Apple //c 128 Ko 2 500,00 F, 384 Ko 3 500,00 F — 1 Écran plat 1 000,00 F — 1 Moniteur + support 600,00 F.
Unité Centrale d'Apple IIG S 512 Ko 3 500,00 F — 1 lecteur 800 Ko 1 500,00 F — Logiciels originaux à débattre.
Joseph Novak
✆ (1) 43.38.92.80 le mardi et mercredi de 18h00 à 20h00.

Vends, pour cause double emploi, unité centrale d'Apple //e avec carte Chat Mauve, faire offre à :
Pierre Vachler
✆ 56.62.21.87

Cherche contacts sur tout Apple, dans le bordelais, pour ne pas évoluer seul.
Pierre Vachler
✆ 56.62.21.87 (entre 19 et 21 h)

Vends Apple //c, moniteur mono Zénith, souris, deuxième lecteur 5,25", imprimante : 6 500,00 F.
Champeyrol Jean-Marc
✆ 43 50 76 06

Vends Apple //e 128 Ko 80 colonnes 65C02, 2 lecteurs + moniteur + interface Epson : 4 000,00 F et logiciels : Mouse Desk 2.0, SuperAppleWorks 1.4 & SuperMacroWorks, ProSel, Copy II+ 1.7 ChessMater 2000, etc.
Zip-Chip 4 Mhz 2 000,00 F — Carte Super Série 500,00 F — Souris et carte 600,00 F — Lecteur Unidisk 3,5" et carte 1 500,00 F — Carte AE RamWorks (80 col. + 1 Mo) 4 000,00 F — Disque Dur Profile 5 Mo + carte 2 500,00 F — Interface // graphique 200,00 F — Carte Z80 CP/M 200,00 F.
Le tout : 13 000,00 F.
Pierre Duchesne
Sous la Haie - 35540 Plerguer
✆99.58.90.38

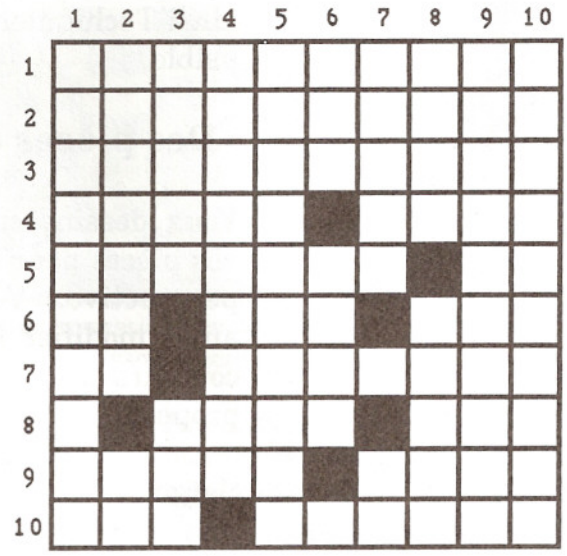
Vends Apple IIGS, écran monochrome 1Mo, HD 20 Mo, 1 lecteur 3,5", VS Com 21 : 10 000,00 F.
Imprimante ImageWriter I grand chariot 1 500,00 F — OMNIS III, Works 1,2" pour Macintosh (avec factures), prix interressant.
Dr Bismuth
✆ 49.09.55.48

Horizontalement

1 - Ensemblier
2 - Pas nécessairement naturiste !
3 - Plaisenterait
4 - Tourmente - Général français
5 - Attions - Infinitif
6 - Règle - Béatifiée - Ainsi de suite
7 - Mais - Classerai
8 - Aïeul - Sa baie est célèbre
9 - Panier percé - Département
10 - Bon copain - Coffres

Verticalement

1 - Effectua un nettoyage !
2 - Isoler - Donc avant le déjeuner
3 - Rendit illisible - Dans l'alphabet
4 - Oxydes de fer
5 - Reproduits
6 - An - Insecte
7 - Traînes - Personnel
8 - Socle renversé - Déménageât
9 - N'est donc pas un gadget !
10 - Rererelabourions



Problème 44
par
Joëlle Piard

Solution du n° 43

P	A	T	I	S	S	E	R	I	E
A	R	R	E	T	E	R	E	N	T
S	T	A	L	I	N	I	S	T	E
S	U	C	E	M	E	N	T	■	N
E	S	■	T	U	G	■	A	E	D
R	■	I	S	L	A	M	I	S	A
I	O	N	■	A	L	E	■	P	I
O	R	D	I	N	A	I	R	E	S
N	E	E	■	T	I	R	E	R	■
S	E	X	U	E	S	■	M	E	R

Certaines nuits ne portent pas conseil...

Lorsqu'au petit matin le problème demeure, il reste la carte Joker. Elle donne un accès privilégié à notre Hotline téléphonique. Gratuite, la carte Joker accompagne chaque abonnement ; un nouvel atout pour les programmes Pom's.

Apple Expo
1J16-1K15

LOGITECH

ScanMan™ for Mac. L'imagination en liberté.

Imaginez: Saisir n'importe quelle image dans votre Mac™ Plus, SE ou II en quelques secondes...

Avec le scanner à main ScanMan, rien n'est plus simple. Choisissez 100, 200, 300 ou 400 dpi, mode trait ou demi-tons... clickez et scannez toute image jusqu'à 106 mm de large, la longueur dépendant de la mémoire disponible.

Vous pouvez modifier la taille des images, les tourner, inverser, éditer et les transférer vers pratiquement toute application — des logiciels de dessin aux programmes de PAO — ou les sauvegarder dans différents formats de fichiers.

Livré avec boîtier interface SCSI, logiciels (application et DA), manuel d'utilisation et naturellement garantie et support technique Logitech, ScanMan met le pouvoir des images entre vos mains.

NOUVEAU: L'application ScanMan version 2.0 vous offre, en plus de nombreuses nouvelles fonctions, un traitement d'image en vrais niveaux de gris permettant une exceptionnelle qualité de reproduction à l'écran et à l'impression.

Pour plus d'informations, contactez
LOGITECH Vente France
Tél.: (1) 34 21 98 88
Fax: (1) 30 37 66 96

LOGITECH S.A.
Siège Européen
Tél.: ++41-21-869 96 56
Fax: ++41-21-869 97 17

Mac est une marque de Apple Computer, Inc.
ScanMan est une marque de Logitech.